General Disclaimer

One or more of the Following Statements may affect this Document

- This document has been reproduced from the best copy furnished by the organizational source. It is being released in the interest of making available as much information as possible.

- This document may contain data, which exceeds the sheet parameters. It was furnished in this condition by the organizational source and is the best copy available.

- This document may contain tone-on-tone or color graphs, charts and/or pictures, which have been reproduced in black and white.

- This document is paginated as submitted by the original source.

- Portions of this document are not fully legible due to the historical nature of some of the material. However, it is the best reproduction available from the original submission.

Produced by the NASA Center for Aerospace Information (CASI)

SKYLAB S191, S192 PROGRAM DESCRIPTIONS

(NASA-CR-152508) SKYLAB S191, S192 PROGRAM DESCRIPTIONS (Computer Sciences Corp.)
424 p HC A18/MF A01 CSCL 22A

N77-26177

Unclas
G3/12 36340

Prepared For

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
Goddard Space Flight Center
Greenbelt, Maryland

CONTRACT NAS 5-11999
Task Assignment 521

FEBRUARY 1976

JUL 1977
RECEIVED
NASA STI FACILITY
INPUT BRANCH

CSC
COMPUTER SCIENCES CORPORATION

SKYLAB S191, S192 PROGRAM DESCRIPTIONS

Prepared For

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION

Goddard Space Flight Center

Greenbelt, Maryland

CONTRACT NAS 5-11999

Task Assignment 521

FEBRUARY 1976

SKYLAB S191, S192 PROGRAM DESCRIPTIONS

Prepared for

GODDARD SPACE FLIGHT CENTER

By

COMPUTER SCIENCES CORPORATION

Under

Contract NAS 5-11999
Task Assignment 521

Prepared by:

M. Mack Date

Approved by:

Dr. D. Y. Stowell Date
Section Manager

TABLE OF CONTENTS

Section 1 - Introduction		1-1
Section 2 - Data Access Programs		2-1
2.1	LIST191	2-1
2.1.1	Main	2-1
2.1.2	FRATIM	2-2
2.1.3	MYERR	2-4
2.1.4	RDH191 (Entry RD191)	2-5
2.1.5	Hardware Requirements	2-7
2.1.6	Core Requirements	2-7
2.1.7	Time Requirements	2-7
2.1.8	Card Input	2-8
2.1.9	Data Card Format	2-8
2.1.10	Output	2-8
2.2	NWT191	2-9
2.2.1	T91	2-9
2.2.2	CONV	2-12
2.2.3	WRITIT	2-13
2.2.4	ZENPTN	2-15
2.2.5	PTING1	2-16
2.2.6	ERROR	2-18
2.2.7	NWT191	2-19
2.2.8	Hardware Requirements	2-22
2.2.9	Core Requirements	2-22
2.2.10	Time Requirements	2-22
2.2.11	Card Input	2-22
2.2.12	Data Card Format	2-22
2.2.13	Output	2-22
2.3	RD192	2-23
2.3.1	R92	2-23
2.3.2	ERR192	2-24
2.3.3	PRINTH	2-25
2.3.4	PRINTSC	2-27
2.3.5	HRMNSC	2-28
2.3.6	RD192	2-30
2.3.7	Hardware Requirements	2-33
2.3.8	Core Requirements	2-33
2.3.9	Time Requirements	2-33
2.3.10	Card Input	2-33
2.3.11	Data Card Format	2-33

TABLE OF CONTENTS (Cont'd)

Section 2 (Cont'd)

2.3.12	Output	2-33
2.4	CLOUD	2-34
2.4.1	Main	2-34
2.4.2	CLOUD	2-37
2.4.3	PLOTT	2-38
2.4.4	CLDWRT	2-40
2.4.5	Hardware Requirements	2-41
2.4.6	Core Requirements	2-41
2.4.7	Time Requirements	2-41
2.4.8	Card Input	2-41
2.4.9	Data Card Format	2-42
2.4.10	Output	2-42
2.5	Interpolation Programs	2-43
2.5.1	CNTH91	2-43
2.5.2	CNTH92	2-45

Section 3 - Subtask 2 Programs 3-1

3.1	Temperature Humidity Profile Program	3-1
3.1.1	TEMPHUMD	3-1
3.1.2	GETSHL	3-5
3.1.3	NCLOUD	3-5
3.1.4	VIEWAN	3-11
3.1.5	WRNCLD	3-15
3.1.6	BLDATTHP	3-18
3.1.7	Hardware Requirements	3-20
3.1.8	Core Requirements	3-20
3.1.9	Time Requirements	3-21
3.1.10	Card Input	3-21
3.1.11	Data Card Format	3-21
3.1.12	Output	3-21
3.2	Utility Routine	3-22
3.2.1	BLKDAT	3-22
3.2.2	Hardware Requirements	3-25
3.2.3	Core Requirements	3-25
3.2.4	Time Requirements	3-25
3.2.5	Card Input	3-25
3.2.6	Data Card Format	3-25
3.2.7	Output	3-25

TABLE OF CONTENTS (Cont'd)

Section 4 - Subtask 5 Programs		4-1
4.1	QUICK	4-1
4.1.1	MAIN	4-1
4.1.2	TEMBRH	4-6
4.1.3	REFLE1	4-8
4.1.4	REFLE2	4-11
4.1.5	REFLE3	4-13
4.1.6	CLUTHK	4-16
4.1.7	DELTA1	4-18
4.1.8	CONTOU	4-20
4.1.9	WRITT	4-23
4.1.10	SPLINE	4-24
4.1.11	Hardware Requirements	4-26
4.1.12	Core Requirements	4-26
4.1.13	Time Requirements	4-26
4.1.14	Card Input	4-26
4.1.15	Data Card Format	4-26
4.1.16	Output	4-26
4.2	QUICK3	4-27
4.2.1	Main	4-27
4.2.2	TEMBRH (Version 2)	4-30
4.2.3	REFLE1 (Version 2)	4-31
4.2.4	CONTOU (Version 2)	4-33
4.2.5	CLUTHK (Version 2)	4-35
4.2.6	Hardware Requirements	4-36
4.2.7	Core Requirements	4-36
4.2.8	Time Requirements	4-37
4.2.9	Card Input	4-37
4.2.10	Data Card Format	4-37
4.2.11	Output	4-37

Appendix A - LIST191 Run Program Listings, Input, and JCL Output

Appendix B - NWT191 Run Program Listings, Input, and JCL Output

Appendix C - RD192 Run Program Listings, Input, and JCL Output

TABLE OF CONTENTS (Cont'd)

Appendix D - Cloud S-191 Plot Program Listings, Input, and JCL Output

Appendix E - Interpolation Programs CNTH191 and CNTH92 Program Listings, Input, and JCL Output

Appendix F - Temperature Humidity Profile Program Program Listings, Input, and JCL Output

Appendix G - Block Data Routine Program Listings, Input, and JCL Output

Appendix H - Quick Program Listings, Input, and JCL Output

Appendix I - Quick3 Program Listings, Input, and Output JCL

Appendix J - JSC-Supplied Tape Formats For S-191 and S-192 Tapes

Appendix K - Recorded Wavelengths Found on S-191 Data Tapes and Contents of Reordered S-191 GSFC Version Tapes

Appendix L - Common Traces

Appendix M - Table of Irradiance Values Used as Basis for Subtask 1 Interpolation Routines

Appendix N - Contents of S-192 Output Array

Appendix O - Model Atmospheres Used as a Basis of Computation of Atmospheric Optical Properties

Bibliography

SECTION 1 - INTRODUCTION

This document describes software developed to assist in analysis of Skylab S191 and S192 experiment data. Experiment S192 measured radiation from the Earth in discrete wavelength bands. These discrete bands were organized into separate telemetry channels. Experiment S192 instrumentation scanned an area of the Earth in each of the specified wavelength bands. Although this instrument recorded data from a conical scan, Johnson Spaceflight Center (JSC) scientists at Houston were supposed to send line-straightened data to Goddard Space Flight Center (GSFC) experimenters. Each channel (wavelength band) scan would contain the intensity of that wavelength band at discretized points on the scan line. A combination of channel scans, covering a region on the Earth's surface, would produce an image of that region similar to a color television image, in that discrete wavelengths were superimposed.

Experiment S191 also measured radiance from the Earth. The difference between experiments S191 and S192 is that experiment S191 examined one "spot" (target area) on the Earth in all wavelengths. This instrument was very similar to the instrument flown in the Nimbus Infrared Interferometer Spectrometer (IRIS) Project. It was astronaut operated, and measured the spectral radiance over a milliradian circular field of view.

Programs to access data from the JSC-supplied tapes are described in Section 2. The nominal data format is given in Appendix J. It should be noted, however, that numerous actual format deviations were encountered. Several of these problems are described in Appendix B. These format deviations required frequent and often significant modifications to the data-accessing programs. The exception was the general S191 list program (LIST191) which was not updated at the principal investigator's request. The most current version of the S192 access program (RD192) is described.

The format of the S191 JSC-supplied tapes was found to be inconvenient for analysis at GSFC because radiances and wavelengths were not given in ascending wavelength order and because values were spread among two separate paired tapes. Program NWT191 reordered the data and placed it in a more easily accessible form on new tapes (one for each paired set sent by JSC). Whenever the S191 format changed, significant changes were required in NWT191. Since many tapes were shipped which utilized a particular earlier format of the S191 tapes, both that form of NWT191 and the most current one are included. The change in wavelengths used which was associated with the format change is reflected in Appendix K.

Software to utilize existing algorithms developed by Dr. B. Conrath (GSFC) was developed in cooperation with H. Powell (PMI) and is described in Section 3. The mathematical basis for those algorithms is described in References 1 and 2. CSC modified the programs developed by Dr. Conrath to accommodate S191 data. Those programs with minor modifications will not be separately documented as they have already become a part of the Goddard library. The newly written subroutines and those with significant modifications are discussed.

Since no data existed in the humidity-associated regions (6 to 7 microns), the humidity inversions were never tested. Since the thermal band data was suspected to be poor for inversions (this suspicion was confirmed (Appendix B)), the development of the temperature portion was never completed.

The programs are described in the expectation that a future project with Skylab-type data may utilize them. In addition to the preparation of the Temperature Humidity Inversion Programs, CSC developed a block data generation routine (BLKDAT) which is also included in this presentation. Also, a separate S191 plot program is described, the output of which was utilized by GSFC scientists.

The last group of programs described in this document are the various S192 plotting programs. These programs utilize the RD192 accessing system to

acquire the data, then calculate such parameters as reflectance, temperature, and optical thickness. Finally, plots of these and alphanumeric equivalents based on these are made. All programs utilize the Wolf Plot Package allowing printer, CalComp, or microfilm output.

SECTION 2 - DATA ACCESS PROGRAMS

2.1 LIST191

2.1.1 MAIN

2.1.1.1 Identification

English Language Title: S191 Tape Content Listing Program

Programming Language: FORTRAN

Analyst's Name and Organization: M. Mack, Computer Sciences Corporation

Programmer's Name and Organization: M. Mack, Computer Sciences Corporation

2.1.1.2 Functional Procedure

This program lists in labeled numbered pages the values found on the S191 tapes produced prior to June 1974.

2.1.1.3 Formal Procedure

MAIN calls subroutine RDH191, which accesses housekeeping data, and the housekeeping data is reordered and listed. The user specifies as input the form in which he desires the independent variable (i.e., either microns of wavelength or cm^{-1} wavenumbers). Subsequently, the subroutine entry RD191 is used to repeatedly access regular data records. Subroutine FRATIM converts the frame time, given in 10^{-4} seconds and converted to hours, minutes, and seconds. Data is then listed by JSC title, i.e., XYZ where X is the filter number, Y is always 0, and Z is the detector used. When an error is encountered, appropriate explanatory messages are printed. If the error was on the expected record size, the program terminates after a dump is executed (CALL ABEND). It should be noted that this dump is suppressed unless appropriate

override data is provided on the job card. When a normal end of file is encountered, a descriptive message is plotted and the program terminates normally.

2.1.1.4 Coding Information

Calling Sequence: N/A

COMMON Storage Used: N/A

Input and Output:

Unit 5 - NTYPE option
= 0, wavelength
= 1, wavenumber

Unit TPS191 - Input 9-track, density 3, unlabeled, JSC-supplied S191 tape with BLKSIZE 2520

Unit 6 - Error, debug, and general message unit on printer

Unit 9 - Tape Content Listing unit on printer

Error Returns: Noted above.

Called Subroutines: ERRSET (system routine), RDH191, RD191, FRATIM, ABEND (system routine)

2.1.1.5 Programs or Subroutines To Use MAIN

N/A

2.1.1.6 References

N/A

2.1.2 FRATIM

2.1.2.1 Identification

English Language Title: Frame Time Conversion Program

Programming Language: FORTRAN

Analyst's Name and Organization: F. Milasi, Computer Sciences Corporation

Programmer's Name and Organization: F. Milasi, Computer Sciences Corporation

2.1.2.2 Functional Procedure

On regular data records, frame time is recorded in 10^{-4} seconds. For ease of understanding, the time is converted to hours, minutes, and seconds.

2.1.2.3 Formal Procedure

The routine utilizes repeated truncations of divisions of the input and subtraction from that input to yield the required units of time.

2.1.2.4 Coding Information

Calling Sequence: CALL FRATIM (IY, ITIME)

where IY is the input frame time in 10^{-4} seconds, and is I*4 data

ITIME is the output array of time in hours, minutes, and seconds, and is I*2 data

COMMON Storage Used: N/A

Input and Output: Through the calling sequence.

Error Returns: N/A

Called Subroutines: N/A

2.1.2.5 Programs or Subroutines To Use FRATIM

MAIN of LIST191 and NWT191.

2.1.2.6 References

N/A

2.1.3 MYERR

2.1.3.1 Identification

English Language Title: Abnormal termination routing routine through ERRSET when a maximum allowable number of system errors is encountered.

Programming Language: FORTRAN

Analyst's Name and Organization: M. Mack, Computer Sciences Corporation

Programmer's Name and Organization: M. Mack, Computer Sciences Corporation

2.1.3.2 Functional Procedure

One option of the system ERRSET routine allows users to control system action after the maximum number of system errors (209-214) are encountered through a control switch to a specified user routine. The routine here, MYERR, thus calls ABEND to yield a dump for ease of debugging.

2.1.3.3 Formal Procedure

The only executable statement in the program is the call to the system routine ABEND.

2.1.3.4 Coding Information

Calling Sequence: CALL MYERR (no arguments)--never actually appears in code; control passed by routine ERRSET.

COMMON Storage Used: N/A

Input and Output:

Unit SYSUDUMP - Requires SYSUDUMP JCL card

Error Returns: N/A

Called Subroutines: N/A

2.1.3.5 Programs or Subroutines To Use MYERR

MAIN of LIST191, T91 (MAIN of NWT191) and R92 (MAIN program of RD192)

2.1.3.6 References

N/A

2.1.4 RDH191 (Entry RD191)

2.1.4.1 Identification

English Language Title: S191 tape accessing program

Programming Language: Assembly Language Code

Analyst's Name and Organization: M. Mack, Computer Sciences Corporation

Programmer's Name and Organization: M. Mack, Computer Sciences Corporation

2.1.4.2 Functional Procedure

This subroutine reads the S191 data tapes, calculates the actual radiance value from its parameterized form, and passes back the values to the calling program for listing.

2.1.4.3 Formal Procedure

The first part of the program, RDH191, which is called once, accesses and unpacks the Housekeeping data record, calculates and stores the data slopes and intercepts, then passes unpacked Housekeeping data to the calling program for listing. Subsequent calls enter through RD191. There, radiance data parameters are accessed and unpacked.

JSC records data in 1-byte parameter values (P) from which, using slopes (S) and intercepts (I), radiance values (R) can be calculated as $R = S * P + I$ (see Appendix A for formats). These values are placed in the array DATH and passed to the calling program for listing. The program was written prior to

tape reception, so it allows for the promised possibility of 2-byte length data. The logic never had to be used. Since more than one "block" of data appears in one physical record, the tape need not be accessed each time a spectrum is accessed, unpacked, modified, and stored. However, a running counter of how many "blocks" have been used and, therefore, a record of where accessing is to commence within the array of physical block contents, DAT, must be kept. This counter is checked upon entry and the record reinitialized whenever a new physical record must be sought.

Finally, variables called BUGX, where X varies between 1 and C are used to aid in tracing errors (should any unforeseen ones arise). These are initialized to 0, but are set to their X value whenever certain operations are performed. Bugs associated with Operations occurring upon repeated entries are reinitialized before control returns to the calling program.

2.1.4.4 Coding Information

Calling Sequence: First entry CALL RDH191 (DATH, NCOND, NTYPE)

Subsequent entries CALL RD191 (DATA, NCOND)

where DATH is the array of tape housekeeping data

NCOND is an indicator supplied by subroutine RDH191
- = 1, housekeeping data
- = 2, regular data
- = 3, end of file
- = 4, error, e.g., variable for radiance parameter was more than 2 bytes long

NTYPE is an indicator supplied by the user
- = 0, wavelength in microns is desired for independent variable
- = 1, wavenumber in $(cm)^{-1}$ is desired for independent variable

DATA is the array of regular radiance data

COMMON Storage Used: N/A

Input and Output:

Unit TPS191 - 9-track; RECFM = U; blocksize = 2520; unlabeled JSC-supplied S191 input tape

Error Returns: If NCOND = 4, some error was encountered. Among the possibilities are: (1) there were more than the promised 1 or 2 bytes per record; (2) the blocksize was too large; (3) the logic for calculating the index value of the input array in the assembly access is incorrect--the wrong value was present. Either the logic was in error or the tape was not as promised.

Called Subroutines: N/A

2.1.4.5 Programs or Subroutines To Use RDH191

MAIN of LIST191.

2.1.4.6 References

N/A

2.1.5 Hardware Requirements

OS/360 tape drive for input tape
OS/360 card reader for deck and NTYPE input data card
OS/360 printer for listings

2.1.6 Core Requirements

The core requirement for the entire program is 200 bytes.

2.1.7 Time Requirements

The time requirement (which varies significantly) is highly dependent upon the size of the tape sent by JSC. At present, it has been found that no tape has required more than 4 minutes CPU and 4 minutes I/O. Most require less than half those figures.

2.1.8 Card Input

One containing a numeral for NTYPE

= 0, wavelength in microns
= 1, wavenumber in cm^{-1}

2.1.9 Data Card Format

Format (I5)

2.1.10 Output

A sample output is given in Appendix A together with a listing, JCL, and input card.

2.2 NWT191

2.2.1 T91

2.2.1.1 Identification

English Language Title: Driver Program for creation of GSFC version of S191 tapes

Programming Language: FORTRAN

Analyst's Name and Organization: M. Mack, Computer Sciences Corporation

Programmer's Name and Organization: M. Mack, Computer Sciences Corporation

2.2.1.2 Functional Procedure

This program reads the appropriate positional angles and serves as the driver for the main tape-creating program, NWT191.

2.2.1.3 Formal Procedure

The desired output tape is read and placed in COMMON TAPEN for the use of subroutine WRITIT. At user option, zenith angle data (extracted by hand from Boresight camera booklets when they are available) and/or pointing angle data (extracted by hand from Boresight camera films when they are available) is read by the program. Then their times are converted to a form compatible with subroutine ZENPTN, which uses them, and all are placed in COMMON VALUES for subsequent use. It should be noted that pointing angles (down, left, up, right) never have more than two values nonzero. Subroutine NWT191 is called to perform the actual tape access and new tape creation. Then, at the end of file, the output tape is unloaded, a descriptive message is written, and the program terminates normally.

2.2.1.4 Coding Information

Calling Sequence: N/A

COMMON Storage Used:

TAPEN contains: TAPE - Real * 8 unit 12 output tape name

VALUES contains: ITIMPT - Associated values for pointing angle data (calendar day, last two digits of year, time into day in 10^{-4} seconds)

PTING - Pointing angle data array

ITIMZN - Associated time values array for zenith angle data (calendar day, last two digits of year, time into day in units of 10^{-4} seconds)

ZENVL - Zenith angle data array

NOPT - Number of groups of pointing angle values (group consists of the four different pointing angles)

NOZE - Number of zenith angle values

ISDTPT - Logical variable indicating whether pointing angle data is available

ISDTZN - Logical variable indicating whether zenith angle data is available

Input and Output:

Unit 5 - N/A

New Card - Tape name (TAPE); format (A8); left adjusted

New Card - Presence of pointing data (ISDTPT); format (L5); left adjusted

New Card - Five time variables (calendar day, year, hour, minute, and second); four pointing angle values; format (I3, 1X, I2, 1X, 3I2, 3X, F2.0, 3X, F2.0, 2X, F2.0, 4X, F2.0)

New Card - (Present whether or not pointing angle data is present.) Termination indicator Δ-9 in columns 1, 2, and 3

New Card - Presence of zenith angle data (ISDTPT); format (L5); left adjusted

New Card - Three time values (hour, minute, and second); fractional part of second; elevation; zenith angle; calendar day; year of data; format (3I2, F4.3, 18X, F4.1, 21X, F4.1, 5X, I3, 2X, I2)

New Card - (Present whether or not zenith angle data is present.) Termination indicator -9 in columns 1 and 2

Error Returns: If NCOND = 2, some error was encountered in processing the tapes. An appropriate error message is printed and a dump, through a call to the system routine ABEND, is requested. As mentioned earlier, this dump is only secured at present if job ID corresponds to the current date (XYY, where X is hexadecimal month and YY is day of month). The job then terminates. In the later included versions, this possibility was eliminated, though some logic remains. Error messages are printed and job terminates whenever any unit 5 input is not compatible in values with expected formats.

Called Subroutines: ERRSET, NWT191, WRITEE (entry to WRITIT), ABEND

2.2.1.5 Programs or Subroutines To Use T91

N/A

2.2.1.6 References

N/A

2.2.2 CONV

2.2.2.1 Identification

English Language Title: Time Conversion Program (month and day to calendar day)

Programming Language: FORTRAN

Analyst's Name and Organization: F. Milasi, Computer Sciences Corporation

Programmer's Name and Organization: F. Milasi, Computer Sciences Corporation

2.2.2.2 Functional Procedure

The routine converts input month and day to calendar day allowing for leap year if necessary.

2.2.2.3 Formal Procedure

This routine tests which month is present and whether it is in a leap year. Then, depending upon the result, it sets the calendar day equal to the total number of days in the preceding months in a standard year plus the day of the month (plus 1 for leap year if month is greater than 2).

2.2.2.4 Coding Information

Calling Sequence: CALL CONV (YY, MM, DD, JUL)

where YY is the I*4 two digit value of the day of the year
MM is the I*4 two digit value of the month of the year
DD is the I*4 two digit value of the day of the month
JUL is the I*4 three digit value of the calendar day

COMMON Storage Used: N/A

Input and Output: N/A

Error Returns: N/A

Called Subroutines: N/A

2.2.2.5 Programs or Subroutines To Use CONV

NWT191

2.2.2.6 References

N/A

2.2.3 WRITIT

2.2.3.1 Identification

English Language Title: Output tape and printer listing producing subroutine for new S191 tape creation program

Programming Language: FORTRAN

Analyst's Name and Organization: M. Mack, Computer Sciences Corporation

Programmer's Name and Organization: M. Mack, Computer Sciences Corporation

2.2.3.2 Functional Procedure

The routine writes the new GSFC version of the S191 data tapes and displays the values on the printer for scientists' personnel.

2.2.3.3 Formal Procedure

On the first call to WRITIT, the output tape is mounted. On subsequent calls, this logic is bypassed. The data array (currently 450 full words long) is then written on the output tape. Next, housekeeping data, such as time and zenith angle, are listed on a unit 9 printer output. Then, by utilizing an auxiliary

output array, and carefully arranging the content, the wavelength and corresponding radiance values are printed side by side, four pairs to a line. Control then returns to the calling program. The second entry, WRITEE, is used to unload the output tape after all spectra formed have been listed.

2.2.3.4 Coding Information

Calling Sequence: (Main)--CALL WRITIT (DATUS, IDATUZ)

(Final)--CALL WRITEE (no arguments)

where DATUS is the 450 word Real * 4 array of output values (housekeeping, wavelengths, and radiances)

IDATU2 is the 900-word Integer * 2 equivalent array of output values (the same array) used for listing the halfword, fixed-time values for the spectrum (in the housekeeping portion)

COMMON Storage Used:

TAPIN: Refer to Section 1.2.1.4

Input and Output:

Unit 12 - Output tape unit

Unit 9 - Printer unit for spectra and housekeeping values

Error Returns: N/A

Called Subroutines: MOUNT, FWRITE, UNLOAD (OS 360 FTIO routines described in GSFC manuals)

2.2.3.5 Programs or Subroutines To Use WRITIT

WRITIT called by NWT191 assembly routine

WRITEE called by T91 (main driver program of NWT191 overall program)

2.2.3.6 References

N/A

2.2.4 ZENPTN

2.2.4.1 Identification

English Language Title: Zenith angle and pointing angles determination-interpolation routine

Programming Language: FORTRAN

Analyst's Name and Organization: F. Milasi, Computer Sciences Corporation

Programmer's Name and Organization: F. Milasi, Computer Sciences Corporation

2.2.4.2 Functional Procedure

ZENPTN utilizes the time arguments passed in the calling sequence to interpolate between data array values passed to it by the Main Driver Program, T91, through COMMON, yielding zenith angle (through a subroutine call) and pointing angle data.

2.2.4.3 Formal Procedure

As described in Section 2.2.1.3, zenith angle and pointing angle data (when available) covering the same time period as is covered by the tapes, is placed in COMMON VALUES by T91. Upon entry to ZENPTN, the availability of zenith data is tested by examination of the logical variable ISDTZN from COMMON. If present, a series of time checks ensures that the time requested is within data time bounds. The interval within which the input time lies is then determined and simple linear interpolation is used to yield the desired zenith value. If no zenith data is present or if input time "fails" one of the various time checks, the zenith angle is set to zero, an error message is printed (through a call to subroutine ERROR), and processing continues.

A similar procedure, accomplished by a call to subroutine PTING1, yields pointing angle values. Control then returns to the calling program.

2.2.4.4 Coding Information

Calling Sequence: CALL ZENPTN (CALDAY, YEAR, FRTM, ZENITH, PTAN)

where CALDAY is the input calendar day of the year in which the data was sensed

YEAR is the input last two digits of the year in which the data was sensed

FRTM is the input frame time value for the data sensed in units of tenths of milliseconds

ZENITH is the output interpolated zenith angle value

PTAN is the output array of interpolated pointing angle values

COMMON Storage Used:

VALUES: Refer to Section 1.2.1.4

Input and Output: N/A

Error Returns: In any error situation, the sought zenith value is set to zero and an appropriate message is printed by subroutine ERROR. Exit from the routine is a conventional exit and does not preclude further processing.

Called Subroutines: ERROR, PTING1

2.2.4.5 Programs or Subroutines To Use ZENPTN

NWT191

2.2.4.6 References

N/A

2.2.5 PTING1

2.2.5.1 Identification

English Language Title: Pointing angles determination-interpolation routine

Programming Language: FORTRAN

Analyst's Name and Organization: F. Milasi, Computer Sciences Corporation

Programmer's Name and Organization: F. Milasi, Computer Sciences Corporation

2.2.5.2 Functional Procedure

PTING1 utilizes the time arguments passed in the calling sequence to interpolate between data array values, passed to it through the COMMON VALUES, yielding pointing angles.

2.2.5.3 Formal Procedure

Upon entry to PTING1, the availability of pointing angle data is tested by examination of the logical variable, ISDTPT, from COMMON. If present, a series of time checks ensures that the time requested is within data time bounds. The interval, within which the input time lies, is then determined and simple linear interpolation is used to yield the desired pointing angles. If no pointing angle data is present, or if input time "fails" one of the various time checks, the pointing angles are set to zero, an error message is printed (through a call to subroutine ERROR), and processing continues. Control then returns to the calling program.

2.2.5.4 Coding Information

Calling Sequence: CALL PTING1 (CALDAY, YEAR, FRTM, PTAN)

where CALDAY is the input calendar day of the year in which the data was sensed

YEAR is the input last two digits of the year in which the data was sensed

FRTM is the input frame time value for the data sensed in units of tenths of milliseconds

PTAN is the output array of interpolated pointing angle values

COMMON Storage Used:

VALUES: Refer to Section 1.2.1.4

Input and Output: N/A

Error Returns: In any of the possible error situations, the sought pointing angles are set to zero and an appropriate message is printed by subroutine ERROR. Exit from the routine is a conventional exit and does not preclude further processing.

Called Subroutines: ERROR

2.2.5.5 Programs or Subroutines To Use PTING1

ZENPTN

2.2.5.6 References

N/A

2.2.6 ERROR

2.2.6.1 Identification

English Language Title: Error message printing routine

Programming Language: FORTRAN

Analyst's Name and Organization: F. Milasi, Computer Sciences Corporation

Programmer's Name and Organization: F. Milasi, Computer Sciences Corporation

2.2.6.2 Functional Procedure

ERROR serves as the problem description vehicle for subroutines ZENPTN and PTING1.

2.2.6.3 Formal Procedure

The arguments include the number related to the type of error which occurred. Upon entry into ERROR, the message printed is determined by the value of that input number. After the message is printed, control returns to the calling program.

2.2.6.4 Coding Information

Calling Sequence: CALL ERROR (IERR, YEAR, CALDAY, FRTM, &IIII)

where IERR is the input number indicating the type of error which occurred

YEAR is the input last two digits of the year in which the data associated with the error was sensed

CALDAY is the input calendar day of the year on which the data associated with the error was sensed

FRTM is the input frame time at which the data associated with the error was sensed in units of tenths of milliseconds

&IIII is the input statement number in the calling program to which control returns upon exit from ERROR

COMMON Storage Used: N/A

Input and Output:

Unit 6 - Printer unit for error messages

Error Returns: N/A

Called Subroutines: N/A

2.2.6.5 Programs or Subroutines To Use ERROR

ZENPTN, PTING1

2.2.6.6 References

N/A

2.2.7 NWT191

2.2.7.1 Identification

English Language Title: Tape creation routine for GSFC versions of S191 tapes

Programming Language: Assembly Language Coding

Analyst's Name and Organization: M. Mack, Computer Sciences Corporation

Programmer's Name and Organization: M. Mack, Computer Sciences Corporation

2.2.7.2 Functional Procedure

NWT191 reads the paired S191 data tapes, unpacks and reorders the data, and calculates radiance values.

2.2.7.3 Formal Procedure

The first part of the program, whose logic is used only once during each execution, opens the tapes and accesses, unpacks, and stores the Housekeeping record. Note that the units on which input tape A or tape B of the paired S191 tapes are mounted is not crucial (i.e., unit B could be mounted for unit A and vice versa). The program will check for certain indicative parameters and will access data accordingly. It calculates the associated slopes and intercepts, then stores them. Subroutine CONV is called to yield equivalent calendar day for input month, day, and year of data. The next part of the program is concerned with the accessing, unpacking, modification, and reordering of radiance data. Since two to three detectors are present per filter and since data often overlaps, more than one detector value for a filter is nonzero. The program will average calculated nonzero values and place the average as the final radiance value. The last portion accesses, unpacks, and modifies the Skylab data (latitude, longitude, etc.). Subroutine FRATIM is called to convert input frame time to actual hour, minute, and second.

Subroutine ZENPTN is called to provide zenith and pointing angle data (if it is available). Finally, subroutine WRITIT is called to list spectrum content and write the new tapes. The program tests to see if data is still available within the current physical record. If so, processing continues; if not, a new physical record is accessed and current access counter is reinitialized.

Inserted BUG values aid debugging if an abnormal termination occurs.

A minimal amount of tape description is given in the ALC routine to enable most changes to be made in the users JCL.

The primary difference in form between version 1 NWT191 and version 2 NWT191 lies in the position and number of wavelengths and associated radiance values. Early tapes had 180 values, six of which were repetitions (174 distinct values). The ordered values vacillate back and forth between tapes and within filters and detectors.

The newer tapes contain 218 distinct wavelengths (some had 217, but current ones are all 218). Although access does vacillate and groups are unordered, the ordering does not involve inter-filter or inter-detector reordering. See Appendix K for wavelengths present on earlier and on current tapes. For simplicity and speed, the new program was a modified version of the old. However, had the second form of tapes been the only type sent, programming logic and approach might have been different.

2.2.7.4 Coding Information

Calling Sequence: CALL NWT191 (NCOND)

where NCOND is an output indicator
= 1, an end of file was encountered
= 2, an error condition was present

COMMON Storage Used: N/A

Input and Output:

TP191A, TP191B - 9-track input data tapes supplied by JSC in pairs for each time period

Error Returns: One error return, originally present, was eliminated from later versions. Because other programs were pressing, some of the logic, not used, was not eliminated.

Called Subroutines: CONV, FRATIM, ZENPTN, WRITIT

2.2.7.5 Programs or Subroutines To Use NWT191

T91, the Main Driver Program for the NWT191 package.

2.2.7.6 References

N/A

2.2.8 Hardware Requirements

Two OS/360 tape drives for input tapes
One OS/360 tape drive for output tape
One OS/360 card reader for deck
OS/360 printer for listings

2.2.9 Core Requirements

The core requirement for the entire program is 200K bytes.

2.2.10 Time Requirements

This figure is highly dependent upon the size of the tapes sent by JSC (which varies significantly) and upon the availability of zenith and pointing angle data. At present, it has been found that average run time is 1 minute of CPU and 1 minute of I/O time.

2.2.11 Card Input

See Section 2.2.1.4.

2.2.12 Data Card Format

See Appendix B.

2.2.13 Output

A sample is given in Appendix B.

2.3 RD192

2.3.1 R92

2.3.1.1 Identification

English Language Title: S192 Driver Program for simple list

Programming Language: FORTRAN

Analyst's Name and Organization: M. Mack, Computer Sciences Corporation

Programmer's Name and Organization: M. Mack, Computer Sciences Corporation

2.3.1.2 Functional Procedure

R92 is used to yield listings of S192 data tapes whenever no other calculations with the data are desired.

2.3.1.3 Formal Procedure

NAMELIST ASK is accessed to determine if Housekeeping data, scan data, or both are to be listed. The logical variables thus input are checked and the values for the RD192 call sequence are set accordingly. The remainder of the program is a continuous looped access through calls to RD192. Exit from this logical loop occurs if either an error is present (NCOND was set to 1), or an end of file was encountered (NCOND was set to 2). In either case, an appropriate message is printed through a call to ERR192 and the program terminates.

2.3.1.4 Coding Information

Calling Sequence: N/A

COMMON Storage Used: N/A

Input and Output:

Unit 5 - NAMELIST ASK card input unit containing logical variables NHOUST and/or NSCANT (default value for these parameters is false, but explicit input of both would ensure clarity of request).

Error Returns: If an error condition was present on the tape, an appropriate message is printed through a call to the routine ERR192; the program then terminates.

Called Subroutines: RD192, ERR192

2.3.1.5 Programs or Subroutines To Use R92

N/A

2.3.1.6 References

N/A

2.3.2 ERR192

2.3.2.1 Identification

English Language Title: Error message printing routine for 192 access programs

Program Language: FORTRAN

Analyst's Name and Organization: M. Mack, Computer Sciences Corporation

Programmer's Name and Organization, M. Mack, Computer Sciences Corporation

2.3.2.2 Functional Procedure

The program lists all condition messages prompted by the accessing of 192 tapes.

2.3.2.3 Formal Procedure

Upon entry to the program, the value of the status indicator, N , is checked. If it is 1, an error was present and an appropriate message is printed; if it is 2, an end of file was encountered and an appropriate message is printed. In either case, the routine returns control to the calling program. It may be utilized by the user in his RD192 utilization programs to list such messages.

2.3.2.4 Coding Information

Calling Sequence: CALL ERR192(N)

where N = 1, a read error was present

= 2, an end of file was encountered

COMMON Storage: N/A

Input and Output:

Unit 6 - Printer unit for error messages

Error Returns: N/A

Called Subroutines: N/A

2.3.2.5 Programs or Subroutines To Use ERR192

R92

2.3.2.6 References

N/A

2.3.3 PRINTH

2.3.3.1 Identification

English Language Title: Housekeeping record printing routine

Programming Language: FORTRAN

Analyst's Name and Organization: M. Mack, Computer Sciences Corporation

Programmer's Name and Organization: M. Mack, Computer Sciences Corporation

2.3.3.2 Functional Procedure

At user option, the contents of the Housekeeping record can be listed using PRINTH.

2.3.3.3 Formal Procedure

The Housekeeping data is passed through the argument list under four different (but location equivalent) names. Since equivalencing cannot be performed on argument list arrays, and since Housekeeping data consists of Real*4, Integer*4, Integer*2, and Logical*1 (EBCDIC) values, the same values were passed under those four data types. Since the values on the tape often corresponded to set meanings, these meanings are printed. All Housekeeping data is listed on four numbered pages together with any necessary explanatory annotations.

2.3.3.4 Coding Information

Calling Sequence: CALL PRINTH (HOUSEK, IHOUS, IHOUS2, LHOUS)

where HOUSEK is the Real*4 array of S192 Housekeeping data

IHOUS is the Integer*4 equivalent array

IHOUS2 is the Integer*2 equivalent array

LHOUS is the Logical*1 equivalent array

COMMON Storage Used: N/A

Input and Output:

Unit 9 - Printer output unit which will contain the listing of Housekeeping data

Error Returns: N/A

Called Subroutines: N/A

2.3.3.5 Programs or Subroutines To Use PRINTH

RD192

2.3.3.6 References

N/A

2.3.4 PRNTSC

2.3.4.1 Identification

English Language Title: Scan data printing routine

Programming Language: FORTRAN

Analyst's Name and Organization: M. Mack, Computer Sciences Corporation

Programmer's Name and Organization: M. Mack, Computer Sciences Corporation

2.3.4.2 Functional Procedure

At user option, the contents of the scan record can be listed using PRNTSC.

2.3.4.3 Formal Procedure

The scan data is passed through the argument list under four different (but location equivalent) names. Since equivalencing cannot be performed on argument list arrays, and since scan data can consist of Real*4, Integer*4, Integer*2, and Logical*1 values, the same values are passed under the four data types. As with the Housekeeping data, some values in the scan array correspond to set messages. These messages, rather than the indicating value, are listed.

Also, a scan array consists of ancillary block data as well as scan radiance data. Ancillary block data remains the same for all channels. Since it would be redundant as well as wasteful of paper to repeatedly print unchanging ancillary block data, the program only lists these values while listing data for the first channel. The ancillary values as well as any explanatory annotations are printed on one page.

Since not all scan values can be listed on any one page (there are a minimum of 1032), the values are printed on several pages (225 values to a page with each page headed by the time value). All printed scan data pages are consecutively numbered.

2.3.4.4 Coding Information

Calling Sequence: CALL PRNTSC (SCAN, ISCAN, ISCAN2, LSCAN)

where SCAN is the Real*4 array of S192 scan data (including all ancillary data

ISCAN is the Integer*4 equivalent array

ISCAN2 is the Integer*2 equivalent array

LSCAN is the Logical*1 equivalent array

COMMON Storage Used: N/A

Input and Output:

Unit 9 - Printer output unit which will contain the listing of all scan data

Error Returns: N/A

Called Subroutines: N/A

2.3.4.5 Programs or Subroutines To Use PRNTSC

RD192

2.3.4.6 References

N/A

2.3.5 HRMNSC

2.3.5.1 Identification

English Language Title: Time conversion from tenths of milliseconds to hour, minute, and second routine

Programming Language: FORTRAN

Analyst's Name and Organization: M. Mack, Computer Sciences Corporation

Programmer's Name and Organization: M. Mack, Computer Sciences Corporation

2.3.5.2 Functional Procedure

The input frame time in tenths of milliseconds (found within the scan array) is converted to hour, minute, and second.

2.3.5.3 Formal Procedure

Using the time conversion equivalents as well as the normal computer truncation, frame time is conveniently converted to hour, minute, and second.

2.3.5.4 Coding Information

Calling Sequence: CALL HRMNSC (ISCAN, ISCAN2)

where ISCAN is the Integer*4 array of scan data (including current Housekeeping data, such as time)

ISCAN2 is the Integer*2 equivalent array

COMMON Storage Used: N/A

Input and Output: N/A

Error Returns: N/A

Called Subroutines: N/A

2.3.5.5 Programs or Subroutines To Use HRMNSC

RD192

2.3.5.6 References

None

2.3.6 RD192

2.3.6.1 Identification

English Language Title: S192 Data Access Program

Programming Language: Assembly Language Coding

Analyst's Name and Organization: M. Mack, Computer Sciences Corporation

Programmer's Name and Organization: M. Mack, Computer Sciences Corporation

2.3.6.2 Functional Procedure

RD192 reads the S192 data types supplied by JSC, unpacks and modifies the data, and passes the array of values back to the calling program.

2.3.6.3 Formal Procedure

Prior to understanding the actual description of RD192 is the need to discuss array set-up reasoning. Since the number of data points on S192 tapes per channel changes frequently and since, therefore, setting up the output array within the RD192 program would have to have been changed each time, RD192 stores values directly into the area indicated on the call program. It is the user's duty to create an array large enough to accommodate the data on his tape.

A second advantage of this form of storage lies in the user's ability to utilize the erroneously shipped conical scan tapes. The arrays were enlarged and extraneous, differing types, or overlapping data, skipped. Compare the S192 line strengthened tape format with the S192 conical scan tape format (Appendix J) in order to understand how this could be done. Upon entry into RD192, argument entries are stored. If it is the first call, Housekeeping data must be read, unpacked, modified when necessary (such as with slopes and intercepts), and stored.

Since JSC word size often varied from word to word, much overall looping was precluded. The access is more "labored" as a result. If the user indicates by the value of NVAL(2) in call sequence that he wishes Housekeeping data listed, PRINTH is called. Regular data is then accessed, (subsequent calls will skip to this point and bypass Housekeeping logic). Since more than one channel of radiance values is stored on one physical record on the input tape, and since users will often use only one channel, only one channel's data is passed per call. Program logic tests whether data is still present in the access array or whether a new access must be made.

Ancillary block data is present only in the physical record containing the first of the recorded channels. Program logic tests if it is the first (by means of the stored displacement value to the slope-intercept array) and unpacks and processes this data first if present. Subroutine HRMNSC is called to update the hour, minute, and second values in the "Housekeeping" part of the regular array (using the ancillary block frame time in 10^{-4} seconds as input). All east longitudes, recorded on the tape, are converted to west longitudes (with which GSFC scientists are most familiar). Then scan data is unpacked and modified (using stored slopes and intercepts), and the calculated scan's pixel radiance is stored into the array of the calling program.

Finally, if the user indicates by the value of the input parameter NVAL(3) that he wishes the scan values printed, PRNTSC is called. All bugs which appear in logic which is reexecuted upon subsequent entries, are reinitialized. The bugs are used for logic tracing in case an error should occur in execution. Control then returns to the calling program.

2.3.6.4 Coding Information

Calling Sequence: CALL RD192 (SCAN, NVAL)

where SCAN is the output scan array consisting of ancillary block values and all radiance data for one channel

NVAL is an input and output integer array in which

NVAL(1) is an output condition indicator
= 0, data returned is regular data
= 1, data returned is erroneous data (some error was encountered)
= 2, an end of file was encountered
NVAL(2) is an input user request indicator
= 0, no Housekeeping data should be printed
= 1, all Housekeeping data should be printed
NVAL(3) is an input user request indicator
= 0, no scan data should be printed
= 1, all scan data should be printed

NOTE: Program logic is such that the user could list some scans and not others so long as the proper value of NVAL(3) was passed each time (e.g., every nth call NVAL(3) is set to 1 or 0; or the first access to NVAL(3) = 1, while subsequent accesses use NVAL(3) = 0).

COMMON Storage Used: N/A

Input and Output:

Unit TP192 - JSC-supplied S192 data tape

Error Returns: If the file number dynamically calculated in the program does not agree with that supplied by JSC on the tape itself, NVAL(1) is set to 2 and control is immediately returned to the call program without reinitializing any values. Users should test for an NVAL(1) value of 2 and, at their discretion, request a dump to analyze data in core at the time and/or secure an IBM dump of their S192 tape.

Called Subroutines: PRINTH, HRMNSC, PRNTSC

2.3.6.5 Programs or Subroutines To Use RD192

R92, QUICK, QUICK3

2.3.6.6 References

N/A

2.3.7 Hardware Requirements

One OS/360 tape drive for input tape

One OS/360 card reader for deck

One OS/360 printer for listings

2.3.8 Core Requirements

The core requirement for the entire program is 200K bytes

2.3.9 Time Requirements

The time requirement is highly dependent upon the size of the tape sent by JSC, as well as the number of pixels or radiance values per channel. Both can vary significantly.

2.3.10 Card Input

For NAMELIST ASK values, see Section 1.3.1.4.

2.3.11 Data Card Format

N/A

2.3.12 Output

One sample given in Appendix C.

2.4 CLOUD

2.4.1 Main

2.4.1.1 Identification

English Language Title: GSFC version S191 tape access and plot driver program

Programming Language: FORTRAN

Analyst's Name and Organization: F. Milasi, Computer Sciences Corporation

Programmer's Name and Organization: F. Milasi, Computer Sciences Corporation

2.4.1.2 Functional Procedure

This program accesses the GSFC versions of the S191 data tapes, selects that data which either lies within input time restrictions, or all other data on the tape, and calls the plot programs.

2.4.1.3 Formal Procedure

Desired time input is read through a NAMELIST. The GSFC versions of the S191 data tapes are read, and data which lies within the input time restrictions or (if none is specified) all data on tape is processed. The processing is performed by the called subroutine CLOUD. When either time limits are exceeded or the tape file is ended, the plot tape is completed, the number of frames plotted is listed, and the program terminates.

2.4.1.4 Coding Information

Calling Sequence: N/A

COMMON Storage Used:

DAT91: DATUS - Real array of accessed S191 values. Note that integer portions are equivalenced to integer names outside of COMMON

CLOUDS - Logical variable indicating whether the plot is to be designated as cloud data or, more generally, as S191 data

NUM - Plotting devices to be used:

1000 printer
0100 SC4020 35 or 16mm camera
0200 SC4020 8 inch camera film
0300 SC4020 8 inch camera and hardcopy
0001 CalComp (30 inch drum)
0002 CalComp (12 inch drum)

or a combination of these (e.g., 1001 would be 30 inch CalComp plus printer)

NUMM - Principal plotter device (everything is scaled to this device). It is set to 4, CalComp, in program

NNN - Number of frames plotted

REF - Real array of reflectance values

TBB - Real array of temperature values

WRS191 - Logical variable indicating whether a listing of data plotted is desired

Input and Output:

Unit 12 - 9-track, density 3, GSFC version of S191 tape (tape generated by NWT191 programs)

Unit 5 - Card input unit--NAMELIST INPUT with possible content:

PLOTER - Real*8 name of plotting device desired

IPRINT - Integer*4 indicator of printer list desired

= 0, printer output is desired
= 1, no printer output is desired

MONTH - Integer*4 month of data desired (any valid nonzero value will indicate that a time request is present but month must match month of data on tape if time limits are used)

IDAY - Integer*4 day of data requested

IYEAR - Integer*4 year of data requested

IHOUR - Integer*4 beginning hour of data requested

MINUTE - Integer*4 beginning minute of data requested

ISEC - Integer*4 beginning second of data requested

IENDHR - Integer*4 ending hour of data requested

IENDMI - Integer*4 ending minute of data requested

IENDSC - Integer*4 ending second of data requested

Unit 6 - Error and status message printer unit

Error Returns: Since such a short period appears upon a tape, if value of month does not agree with month on tape (provided time limits were specified), there is an error. The program prints an explanatory message and terminates. Day and year are similarly checked. If there is a discrepancy, the program also prints an appropriate message and terminates.

Called Subroutines: FREAD (OS/360 FTIO routine), CLOUD (TASK ROUTINE), SETBLK, ENDPLT, NOSTAE, CPRIME, PLOTST (WOLFFPLOT package routines)

2.4.1.5 Programs or Subroutines To Use Main

N/A

2.4.1.6 References

N/A

2.4.2 CLOUD

2.4.2.1 Identification

English Language Title: Reflectance-temperature calculation and plot access program

Programming Language: FORTRAN

Analyst's Name and Organization: F. Milasi, Computer Sciences Corporation

Programmer's Name and Organization: F. Milasi, Computer Sciences Corporation

2.4.2.2 Functional Procedure

Cloud calculates reflectances and temperatures and calls the plotting program to display calculated values.

2.4.2.3 Formal Procedure

The frame counter, NNN, is updated. Then, the program checks the value of the zenith angle; if it is positive (and nonzero), reflectances may be calculated. The formula used is

$$REF = (\pi * RAD)/(\cos(Z) * TEKVAL)$$

where RAD is the radiance value

Z is the zenith angle

TEKVAL is the irradiance value corresponding to the wavelength for which the radiance value was secured

Extreme values are eliminated so the reflectances are forced to lie between 0 and 0.8. Temperature values are calculated for the thermal region using the inverse Planck function:

$$TEM = 14388./\{WAVE * LOG\,[1. + (37413./(\pi * WAVE5 * RAD))]\}$$

where WAVE is the wavelength for which the radiance value was secured

RAD is the radiance value

Extreme values are eliminated so that temperatures are forced to lie between 190 and 310 degrees absolute. The plotting program, PLOTT, is then called. Finally, control returns to the calling program.

2.4.2.4 Coding Information

Calling Sequence: CALL CLOUD (no arguments)

COMMON Storage Used: DAT91; see Section 2.4.1.4.

Input and Output: N/A

Error Returns: N/A

Called Subroutines: PLOTT

2.4.2.5 Programs or Subroutines To Use CLOUD

Main (of S191 plot program), Temphumd (Main program of temperature humidity profile program)

2.4.2.6 References

N/A

2.4.3 PLOTT

2.4.3.1 Identification

English Language Title: Reflectance-temperature plot program

Programming Language: FORTRAN

Analyst's Name and Organization: F. Milasi, Computer Sciences Corporation

Programmer's Name and Organization: F. Milasi, Computer Sciences Corporation

2.4.3.2 Functional Procedure

PLOTT utilizes Wolf Plotting routines to display reflectance and temperature upon requested output units.

2.4.3.3 Formal Procedure

The grid size and limits for reflectances are set using SETGRD and OGRID. Then the reflectances are plotted using PLOTT. Explanatory labels are inserted. Next, the grid size and limits for the temperature graph are set and those values are plotted. Explanatory labels are affixed to the plot.

Parameters to the Wolf program are set to allow the temperature graph to appear above the reflectance graph. Some pertinent Housekeeping values are then printed. These include zenith angle, time (month, day, year, hour, minute, and second), latitude, and longitude. An appropriate overall label is then affixed depending upon the value of the logical variable clouds. If true, "Cloud data XXX" is written; if false, "S191 data XXX" is written. XXX is the value of the frame counter. If desired by the user, subroutine CLDWRT is then called to list the plotted values. Finally, the frame is advanced and control returns to the calling program.

2.4.3.4 Coding Information

Calling Sequence: CALL PLOTT (no arguments)

COMMON Storage Used: DAT91; Refer to Section 2.4.1.4

Input and Output:

WOLFPLOT tape on unit PLOTTAPE

Error Returns: N/A

Called Subroutines: SETGRD, OGRID, PLOT, HORLIN, VERLIN, EDIT, FRMADV (WOLFPLOT routines), INCORE (OS/360 utility routine), CLDWRT

2.4.3.5 Programs or Subroutines To Use PLOTT

CLOUD

2.4.3.6 References

N/A

2.4.4 CLDWRT

2.4.4.1 Identification

English Language Title: Reflectance-temperature value listing program

Programming Language: FORTRAN

Analyst's Name and Organization: F. Milasi, Computer Sciences Corporation

Programmer's Name and Organization: F. Milasi, Computer Sciences Corporation

2.4.4.2 Functional Procedure

CLDWRT lists all calculated reflectance and temperature values with their accompanying pertinent Housekeeping values.

2.4.4.3 Formal Procedure

The frame number with an appropriate label (depending on the value of clouds) is listed. The time, zenith angle, latitude, and longitude are listed. A set of explanatory titles followed by the paired wavelength and calculated reflectance values (four pairs to a line) are listed. Next, the temperature explanatory titles followed by the paired wavelength and calculated temperature values (four pairs to a line) are listed. Control then returns to the calling program.

2.4.4.4 Coding Information

Calling Sequence: CALL CLDWRT (no arguments)

COMMON Storage Used: DAT91; Refer to Section 2.4.1.4

Input and Output:

Unit 6 - Printer unit to contain listings of reflectance and temperature data

Error Returns: N/A

Called Subroutines: N/A

2.4.4.5 Programs or Subroutines To Use CLDWRT

PLOTT

2.4.4.6 References

N/A

2.4.5 Hardware Requirements

One OS/360 tape drive for input GSFC version of S191 data tape
One OS/360 tape drive for output CalComp tape
One OS/360 card reader for card input NAMELIST, INPUT
One OS/360 printer for listings
One SC 4020 for microfilm output (if desired)

2.4.6 Core Requirements

The core requirement for the Cloud system of program is 220K.

2.4.7 Time Requirements

Cloud timing is dependent upon the number of frames to be plotted. It averages about 1-1/2 minutes CPU and 1-1/2 minutes I/O.

2.4.8 Card Input

NAMELIST INPUT is described in Section 2.4.1.4.

2.4.9 Data Card Format

N/A

2.4.10 Output

A sample appears in Appendix D.

2.5 INTERPOLATION PROGRAMS

2.5.1 CNTH91

2.5.1.1 Identification

English Language Title: Irradiance interpolation routine for S191 data

Programming Language: FORTRAN

Analyst's Name and Organization: M. Mack, Computer Sciences Corporation

Programmer's Name and Organization: M. Mack, Computer Sciences Corporation

2.5.1.2 Functional Procedure

Exact irradiance values corresponding to S191 experiment wavelengths are calculated by interpolation of irradiance data supplied by Dr. Thekakara (GSFC).

2.5.1.3 Formal Procedure

The desired wavelength values are read. For those which lie between 0.4 and 0.74 micron, Gaussian interpolation is performed. The formula is

$$IRR = \frac{\sum_{i=N1}^{N2} IR_i * e^{-[(Nu-Nu_o)/SIGMA]^2 * LOG(2)}}{\sum_{i=N1}^{N2} e^{-[(Nu-Nu_o)/SIGMA]^2 * LOG(2)}}$$

where IRR is the irradiance value

IR_i is the Thekakara irradiance value

Nu is the wavelength corresponding to IR_i

Nu_o is the wavelength for which an irradiance value is desired

SIGMA is SIGMAF times the Nu_o value (where SIGMAF is the filter halfwidth at half the maximum)

N1 and N2, the limits of the summation, are the indices corresponding, respectively, to the associated Thekakara wavelength, which is less than the sought wavelength minus sigma, and to the associated Thekakara wavelength, which is greater than the sought wavelength plus sigma

For data lying between 0.75 and 2.5 microns, straight linear interpolation is performed (since the filter "shape" does not affect the irradiance value in this region according to Dr. Curran of GSFC). The equation used is

$$IRR = IR_{j1} + [(IR_{j2} - IR_{j1})/(WV_{j2} - WV_{j1})] (WAVE - WV_{j1})$$

where IRR is the irradiance value

IR_i are Thekakara irradiance values

WV_i are wavelengths associated with those Thekakara values

WAVE is the wavelength for which the irradiance value is desired

j1 and j2 are the indices of the wavelengths (and corresponding irradiance values) which correspond to those wavelengths that "surround" the sought wavelength

After all values are calculated, four copies of wavelength and interpolated irradiance values are punched on IBM cards for use in reflectance calculating programs. Finally, these calculated values are listed.

2.5.1.4 Coding Information

Calling Sequence: N/A

COMMON Storage Used: DATATH: WVTH--the real array of wavelengths associated with Thekakara irradiance values; AIRTH--the real array of Thekakara irradiance values

Input and Output:

Unit 5 - Contains the input desired wavelength values

Unit 7 - Card punch used for card deck output

Unit 6 - General status message printing unit

Unit 9 - Printer unit on which the wavelengths and calculated irradiance values are listed

Error Returns: N/A

Called Subroutines: N/A

NOTE: Program requires a Blockdata routine which is contained on LOADLIB ZBMJMCRB on the OS/360-75. A listing of it appears in Appendix E.

2.5.1.5 Programs or Subroutines To Use CNTH91

N/A

2.5.1.6 References

N/A

2.5.2 CNTH92

2.5.2.1 Identification

English Language Title: Irradiance interpolation routine for S192 data

Programming Language: FORTRAN

Analyst's Name and Organization: M. Mack, Computer Sciences Corporation

Programmer's Name and Organization: M. Mack, Computer Sciences Corporation

2.5.2.2 Functional Procedure

Exact irradiance values corresponding to S192 experiment wavelengths are calculated by interpolation of irradiance data supplied by Dr. Thekakara (GSFC).

2.5.2.3 Formal Procedure

The 20 wavelength values for which data is required are read. In this case, each wavelength is the average of the wavelengths represented in the channel. Since S192 data was filter sensitive, Gaussian interpolation is used for all values. The formula is

$$IRR = \frac{\sum_{i=N1}^{N2} IR_i * e^{-[(Nu-Nu_o)/SIGMA_i]^2*LOG(2)}}{\sum_{i=N1}^{N2} e^{-[(Nu-Nu_o)/SIGMA_i]^2*LOG(2)}}$$

where IRR is the sought irradiance value

IR_i is the Thekakara irradiance value

Nu is the wavelength corresponding to IR_i

Nu_o is the wavelength for which an irradiance value is desired

$SIGMA_i$ is $SIGMAF_i$ times the Nu_o value (where SIGMAF is the filter halfwidth at half the maximum; for S192, this differs for each channel. Its value is approximately equal to the differences in limits for each channel used)

N1 and N2, the limits of the summation, are the indices corresponding, respectively, to the associated Thekakara wavelength, which is less than the sought wavelength minus $SIGMA_i$, and to the associated Thekakara wavelength, which is greater than the sought wavelength plus $SIGMA_i$

After all values are calculated, four copies of wavelength and interpolated irradiance values are punched on IBM cards for use in reflectance calculating programs. Finally, these calculated values are listed.

2.5.2.4 Coding Information

Calling Sequence: N/A

COMMON Storage Used:

DATATH: WRTH - Real array of wavelengths associated with Thekakara irradiance values

AIRTH - Real array of Thekakara irradiance values

Input and Output:

Unit 5 - Contains the input desired wavelength values

Unit 6 - General status message printing unit

Unit 7 - Punch used for card deck output

Unit 9 - Printer unit on which the wavelengths and calculated irradiance values are listed

SECTION 3 - SUBTASK 2 PROGRAMS

3.1 TEMPERATURE HUMIDITY PROFILE PROGRAM

Since most of these programs did not originate with this task, decks and listings are provided but no documentation is included. Interested persons are referred to the Goddard Program Library Information for the "Temperature/Humidity Program 'Number 4' Version."

Those programs peculiar to this task and originating with it are described in the following subsections.

3.1.1 TEMPHUMD

3.1.1.1 Identification

English Language Title: Temperature-humidity cloud presence decision program

Programming Language: FORTRAN

Analyst's Name and Organization: M. Mack, Computer Sciences Corporation

Programmer's Name and Organization: M. Mack, Computer Sciences Corporation

3.1.1.2 Functional Procedure

TEMPHUMD tests the S191 data to determine the presence or non-presence of clouds. The results determine whether or not subroutines CLOUD or NCLOUD are called.

3.1.1.3 Formal Procedure

The program was edited for the last time prior to the reception of the most recent S191 tapes. As such, dimensions, locations of associated wavelengths, and radiance values may have changed. Initially, ERRSET is called to preclude premature termination because of system errors. NAMELIST INPUT is

read to determine WOLFPLOT package initial values and, if desired, time limits within which the S191 data must lie to be utilized. Then, in loop fashion, input tape names are read, mounted, accessed, and processed. This processing begins with time testing (if time limit criteria were provided). Next, the reflectance at 0.74 micron is calculated using

$$\text{REFLEC} = (\text{RADVAL} * \pi)/(\cos(Z) * \text{THEKVL})$$

where REFLEC is the calculated 0.74 micron reflectance

RADVAL is the radiance value for 0.74 micron for the current spectrum

Z is the zenith angle associated with the current spectrum

THEKVL is the irradiance value for 0.74 micron

If this reflectance is less than some limit (at the time of initial analysis, 0.20), there must not be a cloud present, so logical variable cloud 1, initialized to true, is made false.

Next, the temperature at 11.1 microns is calculated using the Inverse Planck function:

$$\text{TEMPVL} = 14388./\{11.1 * \text{LOG}\,[1. + (37413./(\pi * (11.1)^5 * \text{RAD}))]\}$$

where TEMPVL is the temperature at 11.1 microns

RAD is the radiance value at 11.1 microns for the current spectrum

If this temperature is greater than or equal to some limit (at the time of initial analysis, 273 degrees absolute), then there must not be a cloud present, so the logical variable cloud 2, initialized to true, is made false. If both tests were true (cloud 1 and cloud 2 being both true), it is assumed a cloud is present, the cloud counter is incremented, and subroutine CLOUD is called. If both tests were false (cloud 1 and cloud 2 being both false), it is assumed no cloud is present, the "no-cloud" counter is incremented, and subroutine NCLOUD is called.

If tests gave opposite results, it is arbitrarily assumed a cloud was present, the dubious case counter is incremented, the cloud case counter is incremented, and subroutine CLOUD is called. In any case, the number of processed spectra is incremented and a message giving the current counter value, spectrum information, and calculated values is printed (for debugging purposes to verify values given to two differentiating limits).

The next spectrum is read and processing continues. If an end of file or end of requested time period (if that option is used) is encountered, an appropriate message is printed in which the values of all counters are listed, the tape is unloaded, and the program terminates.

3.1.1.4 Coding Information

Calling Sequence: N/A

COMMON Storage Used: None in this form. Note that when it became apparent that production running of the THP would never become a reality, subroutine CLOUD was still modified and further developed for general plotting purposes. In the process, call arguments to cloud were dropped and a COMMON used for parameter passage. The small amount of further development of THP (specifically in the NCLOUD section) was independent of the development of cloud, as a dummy cloud subroutine (with matching arguments) was used. If the THP is ever revived, call arguments to both CLOUD and NCLOUD will have to be dropped and a COMMON DAT91 (described elsewhere) inserted into both TEMPHUMD and NCLOUD.

Input and Output:

Unit 5 - Needed for input tape names (regular unit 5 access of alphanumeric, left-adjusted, A8 tape volume-serial number). Also

needed for input NAMELIST INPUT which follows all tape names, and contains one or more of the following parameters:

PLOTER - The alphanumeric name of the ploter device desired (CalComp, SC4020FH, SC4020H, SC4020F, Printer)

IPRINT - Option indicating if printer output is desired

= 0, no
= 1, yes

MONTH - Month of data; if zero (initialization value), time limit option is not desired

IDAY - Day of data

IYEAR - Year of data

IHOUR - Beginning hour of data

MINUTE - Beginning minute of data

ISEC - Beginning second of data

IENDHR - Ending hour of data

IENDMI - Ending minute of data

IENDSC - Ending second of data

These are specified only if time limit option is desired. Note that after NAMELIST was read, unit 5 was rewound to access tape names. Execution terminates when value "&INPUT" is sensed.

Unit 6 - Condition or status message printer unit

Unit 12 - 9-track, GSFC version of S191 data tape

Error Returns: N/A

Called Subroutines: ERRSET (system routine to prevent too premature exit because of system errors), NOSTAE, CPRIME, PLOTST, SETBLK, ENDPLT (the last five of which are WOLFPLOT Package routines), MOUNT, FREAD, UNLOAD (the last three of which are OS/360 FTIO routines), CLOUD, NCLOUD

3.1.1.5 Programs or Subroutines To Use TEMPHUMD

N/A

3.1.1.6 References

N/A

3.1.2 GETSHL

This routine was never developed. A value for the shelter temperature was inserted through the Block Data Routine. The objective was an interpolation between input latitude-longitude time and shelter temperatures for various areas of the globe (i.e., for specific time periods for each latitude-longitude cell, a shelter temperature would exist--the routine would utilize time, latitude, and longitude of the spectrum to determine which shelter temperature applied).

3.1.3 NCLOUD

3.1.3.1 Identification

English Language Title: Temperature humidity inversion driver routine

Programming Language: FORTRAN

Analyst's Name and Organization: M. Mack, Computer Sciences Corporation

Programmer's Name and Organization: M. Mack, Computer Sciences Corporation

3.1.3.2 Functional Procedure

NCLOUD calculates the mixing ratio (humidity) and temperatures based on radiative transfer theory using the input spectral radiances.

3.1.3.3 Formal Procedure

On the first entry to the program, the values of pressure at the various levels from the top of the atmosphere to the ground are set. The particular radiance values to be used for the inversion are placed in the smaller array, XI. This includes the window value at 11.1 microns. Next, the shelter temperature is secured (this routine was never developed due to the suspicions of the project scientist about the dubious quality of the S191 data for inversion purposes).

The average value of the change in the log of the pressure from level to level is calculated for later use in TRANSW and TRANSC (transmission subroutines from program S00091 cited above). The viewing angle VWAN is calculated. The secant of the viewing angle is used in ESTH20 (the water vapor computation routine) and in WRNCLD (the listing and display routine. It is also used in TRANSW and TRANSC to correct the values of the transmission coefficients.

The transmission subroutine TRANSC entry, TRNSIN, is called to calculate pressure exponential values used later on TRANSC. Next, the Planck radiance function is utilized to calculate expected radiance for the 11.1 micron wavelength and current shelter temperature. The difference between that and the actual "window" value is used to determine if processing should take place and, if so, if a hotground correction is needed. The determining limits were set by Dr. Curran (GSFC) but may need to be changed if the program is utilized in the future. Then the equivalent brightness temperature, TB, for the window is calculated using the inverse Planck function. This will be used in later calculations for the corrections to radiance values.

The iteration loop is entered in which the temperature and mixing ratio final calculations are performed. All but the first pass reinitialize the transmission

coefficients array for H20. The radiative transfer routine for calculating transmission coefficients as a function of pressure level for C02, TRANSC, is then called. The transmission coefficients for H20 for all levels for temperature associated values are then modified by the transmission coefficients for C02.

A hotground correction for radiance values, XI, is accomplished by subtracting from the given radiance values the difference between the radiance value calculated at the surface using the TB value mentioned above, and the radiance value associated with the shelter temperature at the surface multiplied by the transmission coefficient for H20 at the surface, or

$$XI_i = XI_i - (FSB(XLAM_i, TB) - FSB(XLAM_i, TS)) * TAU_{in}$$

where FSB is the Planck function yielding radiance values when wavelength and temperature are input

$XLAM_i$ is the wavelength associated with XI_i

XI_i is the particular radiance value to be corrected

TB is the equivalent brightness temperature for the window

TS is the current shelter temperature

TAU_{in} is the transmission coefficient for H20 associated with the wavelength $XLAM_i$ at the surface

When it was apparent that at least some (later found to be all) humidity associated radiance values would be zero, logic was inserted to bypass the calculation of water vapor. (Zero radiance values result in system errors.)

UTOTAL, calculated in ESTH20, is the total amount of water vapor present in the atmosphere. Often, convergence of the calculations occurs prior to the five iterations allowed. This is revealed through the relative change in UTOTAL between a prior iteration and the current one. If less than a convergence test value (set in data), iteration ceases. If the iteration loop is complete and no

UTOTAL estimate is made using mixing ratio values, either "first guess" values initially set in block data (if no prior spectrum had non-zero values), or those previously calculated with an earlier spectrum. Finally, calculated results are listed and displayed utilizing subroutine WRNCLD.

3.1.3.4 Coding Information

Calling Sequence: Call NCLOUD (DAT191, IDT191)

where DAT191 is the real array of Housekeeping and spectral radiances
IDT191 is the integer equivalent array (to DAT191)

Note that under TEMPHUMD, this might be changed if a COMMON for DATA, present in the most current version of CLOUD, is utilized.

COMMON Storage Used:

CHA: RMSTMP - Root mean square of temperature calculated in ESTLSQ
RMSH20 - Root mean square of mixing ratio calculated in ESTH20

CO2: TAUCO2 - Real array of transmission coefficients associated with CO2 for all wavelengths under consideration for all pressure levels

CO2 - Real array of the CO2 absorption coefficients to be used in the polynomial fit to the transmission equation as a function of altitude for all wavelengths (used in TRANSC)

XK - Transmission correction factors used to modify the pressure values in the TRNSIN entry to TRANSC

XKMULT - Transmission correction factors used to modify results of the transmission equation calculations in TRANSC

H2O: TO - Real array of first guess temperature values for each level of pressure

QO - Real array of mixing ratio values calculated for each level of pressure

TAU - Real array of transmission coefficients associated with H20 for all wavelengths under consideration for all pressure levels

WATER - Real array of H20 absorption coefficients to be used in the polynomial fit to the transmission equation as a function of altitude for all wavelength (used in TRANSW)

XXX: TS - Shelter temperature (presently input through block data, but ultimately to be calculated for the individual spectrum through interpolation of world-wide, time-wide values)

PS - Pressure at the surface

PT - Pressure at the top of the atmosphere

DH - Average change in the log of the pressures with a change in level

N - Number of boundaries of levels of pressure (surface to top of atmosphere inclusive)

NL - Number of levels of pressure considered

R - Displacement for the array H in ESTLSQ before it is inverted by subroutine MATINT or MATINH (called MATINV in S00091). Note that two inversion routines, one for temperature (MATINT) and one for humidity (MATINH), were created from the one MATINV routine in S00091 since, as coded, it was dimension and storage dependent. On S00091 programs, these were the same for humidity and temperature. Since they differ, two routines allowing different storage were found to be necessary. Doing this was faster than recoding the entire inversion routine

or looking elsewhere (and was practical in view of the likelihood of non-use due to poor S191 data quality).

MTOTAL - Number of wavelengths associated with temperature calculations only

P - Real array of values of pressure at all levels from the top of the atmosphere to the surface

T1 - Real array of calculated temperature values for each level of pressure

HOTGRD - Logical indicator of the need for hotground correction of radiance values

Input and Output:

Unit 6 - Used as printer listing unit for status messages and intermediate processing values

Error Returns: N/A

Called Subroutines: GETSHL, VIEWAN, WRNCLD, TRNSIN (entry to TRANSC), TRANSW, TRANSC, ESTLSQ, ESTH2O (the last five of which are from S00091 with minor or no changes made)

3.1.3.5 Programs or Subroutines To Use NCLOUD

TEMPHUMD

3.1.3.6 References

Goddard Space Flight Center, N73-11588, A Review of Nonstatistical Techniques for the Estimation of Vertical Atmospheric Structure From Remote Infrared Measurements, Drs. B. J. Corrath and I. Revah

3.1.4 VIEWAN

3.1.4.1 Identification

English Language Title: Routine to resolve pointing angles into view angle

Programming Language: FORTRAN

Analyst's Name and Organization: M. Mack, Computer Sciences Corporation

Programmer's Name and Organization: M. Mack, Computer Sciences Corporation

3.1.4.2 Functional Procedure

VIEWAN resolves the four pointing angles found in the Housekeeping part of the array on the GSFC version tapes into a single nadir view angle.

3.1.4.3 Formal Procedure

Only two of the four pointing angles will ever be nonzero (otherwise they could not be recorded down, left, up, or right). Their configuration would be as in Figures 3-1 and 3-2.

From Napier's triangle (spherical trigonometry):

$$\cos \beta = \tan O_2 \tan (\pi/2 - x) = \tan O_2 \cot x$$

and

$$\cos \alpha = \tan O_1 \tan (\pi/2 - x) = \tan O_1 \cot x$$

Now

$$\sin \alpha = \sin (\pi/2 - \beta) = \cos \beta$$

$$\sin \beta = \sin (\pi/2 - \beta) = \cos \alpha$$

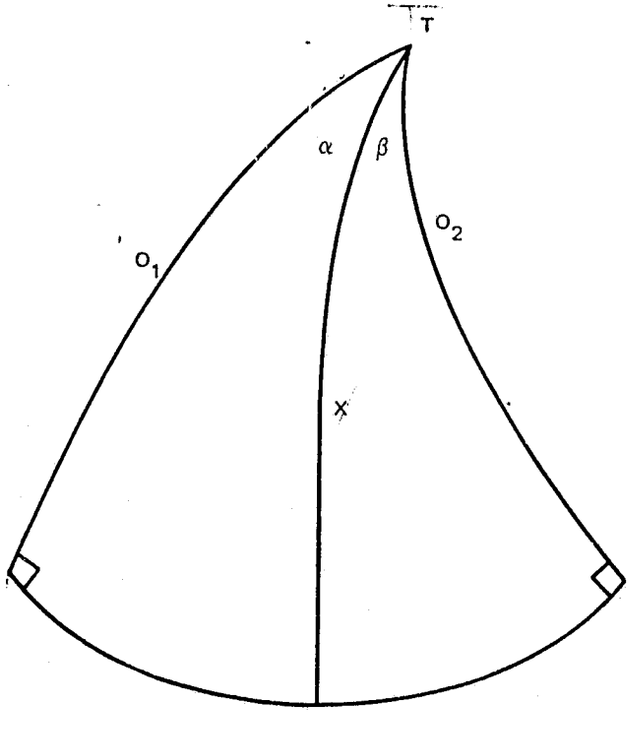


Figure 3-1. Configuration of Pointing Angles

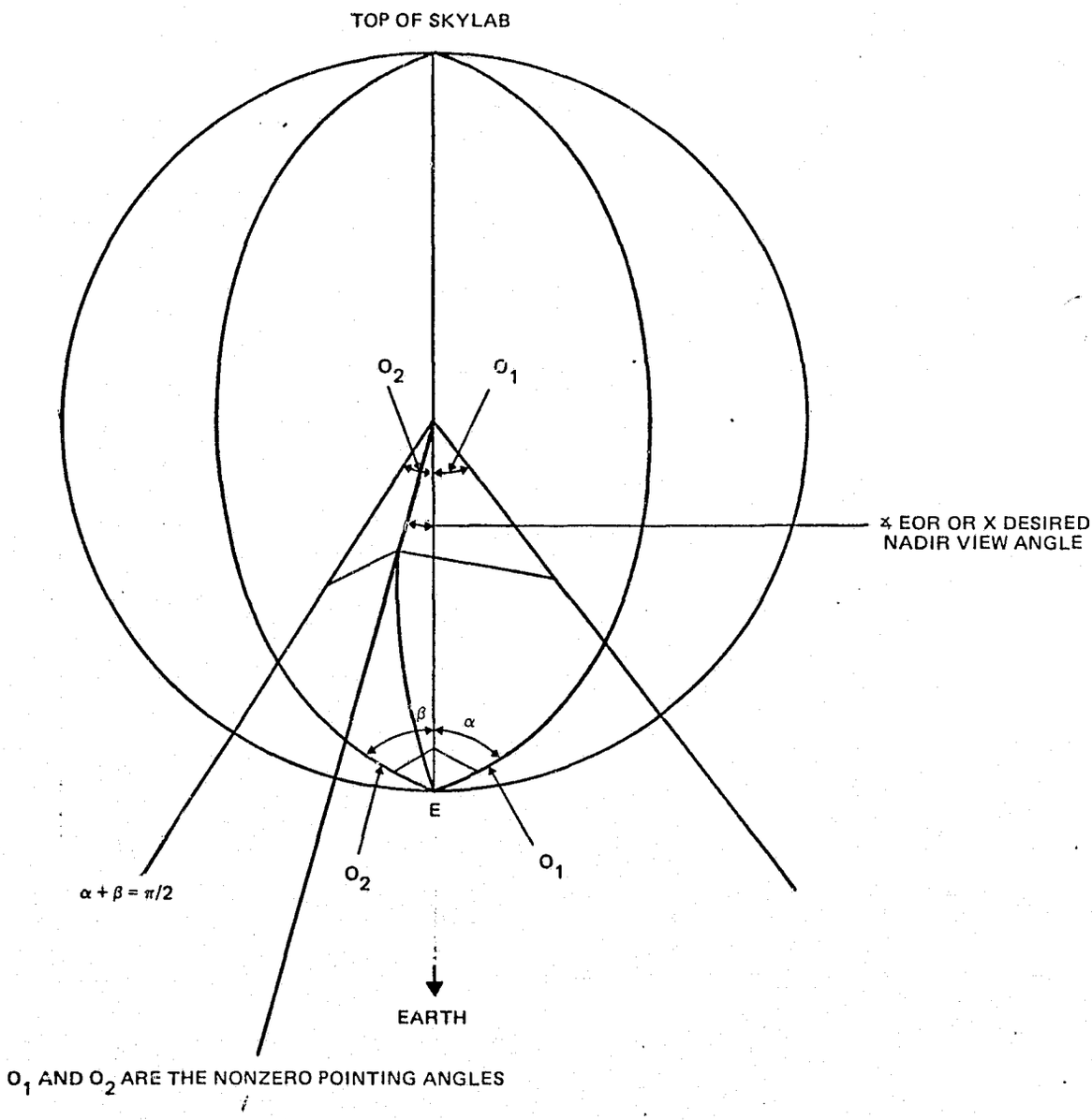


Figure 3-2. Expanded Configuration of Pointing Angles

So, too,

$$\cos \alpha = \tan O_1 \cot x$$

$$\cos \beta = \tan O_2 \cot x$$

Since

$$\sin^2 \alpha + \cos^2 \alpha = 1$$

then

$$(\tan O_2 \cot x)^2 + (\tan O_1 \cot x)^2 = 1$$

or

$$\cot^2 x \, (\tan^2 O_1 + \tan^2 O_2) = 1$$

or

$$\cot^2 x = \left(\tan^2 O_1 + \tan^2 O_2\right)^{-1}$$

or

$$\cot x = \left(\tan^2 O_1 + \tan^2 O_2\right)^{-1/2}$$

Then

$$\tan x = \left(\tan^2 O_1 + \tan^2 O_2\right)^{1/2}$$

or

$$x = \tan^{-1}\left[\left(\tan^2 O_1 + \tan^2 O_2\right)^{1/2}\right]$$

where x is the desired nadir view angle and O_1 and O_2 are the two nonzero pointing angles.

3.1.4.4 Coding Procedure

Calling Sequence: CALL VIEWAN (DAT191, VWAN)

where DAT191 is the input real array of Housekeeping and spectral data

VWAN is the real output nadir view angle

COMMON Storage Used: N/A

Input and Output: N/A

Error Returns: N/A

Called Subroutines: N/A (Only standard FORTRAN compiler trigonometric functions, inverse trigonometric functions, and square root.)

3.1.4.5 Programs or Subroutines To Use VIEWAN

NCLOUD

3.1.4.6 References

N/A

3.1.5 WRNCLD

3.1.5.1 Identification

English Language Title: List and display routines for calculated temperature and humidity values

Programming Language: FORTRAN

Analyst's Name and Organization: M. Mack, Computer Sciences Corporation

Programmer's Name and Organization: M. Mack, Computer Sciences Corporation

3.1.5.2 Functional Procedure

WRNCLD lists and displays the calculated temperature and humidity profile values derived from NCLOUD and its associated radiative transfer subroutines. It also generates output tapes of the values for subsequent analysis.

3.1.5.3 Formal Procedure

Since CPU and IO time is shortened with fewer arguments in the list, the first portion of the routine reorders and places data into an output array. The 9-track output tape is then created. Next, the printer listing is made with inserted explanatory titles and units. The WOLFPLOT package is utilized to place plots of calculated data on the CalComp or SC4020 (as requested by user).

The Housekeeping data is first placed in the Wolf output array through EDIT. Then two loops, one for temperature and one for humidity, place calculated values in the remainder of the array (again using the Wolf EDIT routine). Finally, one last loop plots the output array line by line using the Wolf routine HORLIN. The frame is advanced through FRMADV and control returns to the calling program. Note that since the temperature humidity program was never completed, this routine was never satisfactorily tested. The listing portion appeared to be adequate, but WOLFPLOTs were never made. Subsequent analysts wishing to utilize this program should be cognizant of this.

3.1.5.4 Coding Procedure

Calling Sequence: CALL WRNCLD (XLAM, XI, T1, RH20, UTOTAL, TS, DAT191, IDT191, VIEWAN, ZEROVL)

where XLAM is the real array of all wavelengths associated with temperature or humidity calculations

XI is the real array of temperature- or humidity-related radiances

T1 is the real array of calculated temperatures for all pressure levels

RH20 is the real array of calculated mixing ratios (humidity) for all pressure levels

UTOTAL is the calculated, (or estimated), value of the total amount of water vapor in the atmosphere

TS is the shelter temperature

DAT191 is the real array of Housekeeping and spectrum values

IDT191 is the integer equivalent array of data

VIEWAN is the real nadir view angle

ZEROVL is the logical indicator of whether the humidity inversions were not performed and UTOTAL estimated instead of calculated

COMMON Storage Used: N/A

Input and Output:

Unit 9 - Output printer unit for listing calculated temperature and humidity

Unit 10 - Output 9-track tape unit for calculated temperatures and humidities

PLOTTAPE - Output 7-track WOLFPLOT tape

Error Returns: N/A

Called Subroutines: SAMSIZ, EDIT, HORLIN, FRMADV (all of which are WOLFPLOT routines)

3.1.5.5 Programs or Subroutines To Use WRNCLD

NCLOUD

3.1.5.6 References

N/A

3.1.6 BLDATTHP

3.1.6.1 Identification

English Language Title: Block data routine for temperature humidity program

Program Language: FORTRAN

Analyst's Name and Organization: M. Mack, Computer Sciences Corporation

Programmer's Name and Organization: M. Mack, Computer Sciences Corporation

3.1.6.2 Functional Procedure

The routine establishes the data values used throughout the radiative transfer routines in the temperature humidity program.

3.1.6.3 Formal Procedure

No programming description would be appropriate here. Note how some of the data was derived in case new values are ever inserted. Specifically, the initial values for TO and QO (the latter called A0 and A1), used in COMMON H2O, were derived through interpolation between values for model atmospheres for midlatitude summer (most appropriate for the data we first considered). Midlatitude winter values are also available (Appendix O). More accurate values for the longitude band, etc., might enhance convergence and accuracy but were deemed unnecessary by Dr. Curran (GSFC). The transmission coefficients for H2O and CO2 were derived through interpolation between empirical absorption

coefficients for water vapor and for CO2 found in an ESSA (NOAA) technical report by W. L. Smith of the National Environmental Satellite Center (Appendix O).

Since the wavelengths associated with humidity were so distant from those present in the tables, the calculated values for those wavelengths are not accurate. However, as our brief utilization of the program never involved humidity calculations, the derivation of more accurate values was never considered necessary. Future users, however, should seek more accurate values. The R displacement value was set on the basis of several runs, made to determine proper convergence and behavior of output. In each, R was given a different value. The current one, 4.E-11, appeared most desirable to Dr. Curran (GSFC).

3.1.6.4 Coding Procedure

Calling Sequence: N/A

COMMON Storage Used:

H2O--Refer to Section 3.1.3.4.

SHLDT--TSVAL: Shelter temperature values for all areas for all times

TIMES: Times associated with those shelter temperatures

XXX--Refer to Section 3.1.3.4.

CO2--Refer to Section 3.1.3.4.

ZZZ--XIO: Array of radiance values used together with current radiance values to calculate RMS residual/radiances. Initially set to zero since at the time, project scientists were not interested in this RMS. Probably should be first guess values for radiances for those wavelengths.

PHI: Storage array for kernel matrix in ESTLSQ

Note that COMMON ZZZ was left from S00091. If program had been initiated with Skylab, a COMMON would not have been used here.

Input and Output: N/A

Error Returns: N/A

Called Subroutines: N/A

3.1.6.5 Programs or Subroutines To Use BLDATTHP

All radiative transfer programs referencing COMMONS.

3.1.6.6 References

1. Goddard Space Flight Center, N73-11588, A Review of Nonstatistical Techniques for the Estimation of Vertical Atmospheric Structure From Remote Infrared Measurements, Drs. B. J. Conrath and I. Revah

2. National Environmental Satellite Center, NESC47, A Polynomial Representation of Carbon Dioxide and Water Vapor Transmission, W. L. Smith, February 1969

3.1.7 Hardware Requirements

One OS/360 card reader for input

Three OS/360 tape drives

- Two 9-track: For input GSFC version S191 tape and GSFC output temperature humidity profile tape
- One 7-track: WOLFPLOT tape

One OS/360 printer unit

One SC4020 microfilm processor

3.1.8 Core Requirements

The core requirement for the entire program is less than 450 bytes.

3.1.9 Time Requirements

As with most Skylab programs, the time requirement is highly dependent upon the size of the GSFC S191 version tapes created (which depends upon the size of JSC-supplied tapes). At present, no run took more than 2 minutes CPU and 3 minutes IO. However, no production runs were made and no humidity calculation logic was utilized.

3.1.10 Card Input

See Appendix F.

3.1.11 Data Card Format

See Appendix F.

3.1.12 Output

See Appendix F.

3.2 UTILITY ROUTINE

3.2.1 BLKDAT

3.2.1.1 Identification

English Language Description: Block data creation routine

Programming Language: FORTRAN

Analyst's Name and Organization: M. Mack, Computer Sciences Corporation

Programmer's Name and Organization: M. Mack, Computer Sciences Corporation

3.2.1.2 Functional Procedure

BLKDAT will create a block data routine of data input to it, setting up all required sub-arrays and equivalences.

3.2.1.3 Formal Procedure

The biggest problem the user had (before this routine was developed) was the restrictions to 15 construction cards for all data. For large arrays, this required hand calculations of equivalency positions. This routine obviates that necessity. Since the output format desired by the user must be placed in Logical*1 format "arrays" and not always in positions convenient for Real*8 equivalencing, the input format is placed in another Real*8 word with a Logical*1 array equivalenced to it. Next, the number of elements on a line (no more than 64 allowed--columns 7 through 70) is calculated. Note that division is by one greater than the input format size to allow for commas.

The number of lines required to display the entire array is then calculated. The number of arrays which will have the same number of elements and the number or total are calculated. Next, the number of elements in each array of the same size and the number in the list array are calculated. Only two equivalences

could fit on a line, so the question of whether an odd number of equivalenced arrays (needing a different output format for the last line) is solved through calculation. The data type on the dimension statement should be listed, so indices of the type array which apply are set.

All output of the program is in pairs: unit 7 (card punch) and unit 6 (printer listing). First, the words "block data" and the array with type and dimension are listed and punched. The rest of the program is divided by type. The first division is between real and integer. Later, in the case of real, between Real*4 and Real*8. For integers, the logic divides into Integer*2 and Integer*4. In each case, the output formats established for that type are modified by the insertion of input and calculated values (e.g., number of values to a line).

For debugging purposes, the generated formats are listed on a different printer unit (unit 9). The equivalenced arrays are output dimensioned under the appropriate type of statement and the equivalence statement is also generated. Next, the data, with appropriate equivalenced name preceding it, is output. Finally the "Return" and "End" statements are generated.

The program allows as many as 100 arrays for equivalencing. For most cases, this should be sufficient to accommodate data.

3.2.1.4 Coding Information

Calling Sequence: CALL BLKDAT (NTYPE, ARR4, ARR8, INTAR2, INTAR4, NOELEM, ARNAME, FMT, IFMTSZ)

where NTYPE is an indicator of data type
= 1, Real*4
= 2, Real*8
= 3, Integer*2
= 4, Integer*4

ARR4 is the Real*4 array which contains data if NTYPE = 1; otherwise, it is a single-element, Real*4 dummy array

ARR8 is the Real*8 array which contains the data if NTYPE = 2; otherwise, it is a single-element, Real*8 dummy array

INTAR2 is the Integer*2 array which contains the data if NTYPE = 3; otherwise, it is a single-element, Integer*2 dummy array

INTAR4 is the Integer*4 array which contains the data if NTYPE = 4; otherwise, it is a single-element, Integer*4 dummy array

NOELEM is the dimension of the input array (the number of elements in it)

ARNAME[1] is the Real*8 alphanumeric name of the input array

FMT is the desired data format: IN, EN.I, DN.I, FN.I, AN (with standard FORTRAN meaning)

IFMTSZ has the value N in the above examples (it is the width of the desired format)

COMMON Storage Used: N/A

Input and Output:

Unit 6 - Printer unit listing generated block data routine

Unit 7 - Punch and unit containing generated block data routine in card deck form

Unit 9 - Printer unit containing debug information, i.e., generated format, etc. These references may be removed or a dummy unit inserted as desired by user

Error Returns: N/A

Called Subroutines: INCORE (system routine)

3.2.1.5 Programs or Subroutines To Use BLKDAT

N/A

3.2.1.6 References

N/A

[1] Note that the only naming restriction on the output array is thus necessitated by the establishment of equivalencing array names, i.e., A1 through A100 and I1 through I100. These names cannot be used.

3.2.2 Hardware Requirements

OS/360 card punch

OS/360 printer unit

3.2.3 Core Requirements

The core requirement for the entire test program is 100K bytes. (BLKDAT, without printer or punch allowances, is 8288 bytes.)

3.2.4 Time Requirements

In the test case, time requirements were less than 1/2 minute CPU and 1 minute IO.

3.2.5 Card Input

N/A

3.2.6 Data Card Format

N/A

3.2.7 Output

See Appendix G.

SECTION 4 - SUBTASK 5 PROGRAMS

4.1 QUICK

4.1.1 MAIN

4.1.1.1 Identification

English Language Title: S192 quick-look plot driver program

Programming Language: FORTRAN

Analyst's Name and Organization: F. Milasi, Computer Sciences Corporation

Programmer's Name and Organization: F. Milasi, Computer Sciences Corporation

4.1.1.2 Functional Procedure

This program, through subroutine calls and a minimal amount of calculation, accesses S192 data tapes and plots every 30th point on every 30th line for a fast overview of the entire scene.

4.1.1.3 Formal Procedure

Through NAMELIST INPUT, user requests for channels to be printed, output units to be used, number of points to a line and types of plots desired are recorded. Channels are read by the actual channel number (not its order of appearance on tape since many are missing). The first, second, and fourth channels will be used for reflectance, and the third channel, for temperature. Since JSC sent some conical scan tapes instead of line straightened data (as promised), logic was inserted to utilize the previously developed RD192 program but to bypass dissimilarities present in conical scan data (see Appendix J for further information).

On the conical tapes, additional records are present. The first 18 of every group of scan radiance data must be ignored. Since there are two scan radiance

data groups placed contiguously in the physical block, the first scan must bypass 18 pixels while the second must bypass 36 pixels. To achieve a uniform size of 1032, the last pixel used in the first group would be 1050, while the last pixel used in the second group would be 1068. Since several physical blocks are read and since this is true for every one, an even or odd determination is made by calculating count of the scan mod 2.

The tape is then accessed and the channel number present is compared with the input request list. If present, control shifts to the first, second, third, or fourth channel processing section. If it is none of them, a new scan is accessed. The four processing sections are similarly structured. The line number is accessed, count of passes into that logic section is incremented, and row mod number is calculated. NUMLIN is the skip factor (usually 30).

If the moding result is not 1, this scan is skipped and a new one accessed. The row number is assigned by calculating

$$NROW2 = LINEN2/NUMLIN + 1$$

where LINEN2 is the actual line number and NUMLIN is the skip factor.

The selected subset of the data is then placed into storage arrays, (A for the first channel, B for the second, etc.). For channels 1 and 2, additional processing is performed: the zenith angle is calculated, converted to radians, and recorded. Checks are present to prevent accidental over-storage, i.e., since storage arrays are 35 by 67, no storing is performed if either value is exceeded.

The current version includes two column checks (the first of which precludes the second), necessitated by the present paper shortage reaction measures. Under the current printer line density increase, first the 35th and more recently the 34th pixel would not appear on the line. The working sign of the current arrays is 33 by 67 (logic was left unchanged other than this in the hope that

future paper supplies having increased, the printer output size will be restored). Whenever the working row number is 67, logic shifts to the display of data (since all arrays are clearly filled). If the NUMPLT value indicates that no plot other than printer is required, WOLFPLOT calls are bypassed. In any case, if a third channel is present, brightness temperature is calculated and plotted (as requested) through a call to TEMBRH. If channel 1 is present, reflectance is calculated and plotted through a call to REFLE1 (on which the second and fourth channels, if present, are also processed).

The program then terminates. If an end of file was encountered prior to the completion of the 35 by 67 array, the current counter values are listed as are the requests from the NAMELIST. The plotting and/or listing is then performed on the smaller subset, (i.e., brightness temperature is calculated and printed, if channel 3 is present, through TEMBRH, and reflectance is calculated and printed through REFLE1).

4.1.1.4 Coding Information

Calling Sequence: N/A

COMMON Storage Used:

VALUES: R1 - Real array of reflectance calculated from channel 1

T1 - Real array of optical thickness calculated from R1

RR - Real array equal to X1 array values times 100 (for printer contour)

Y1 - Real array of ratio of reflectances calculated using T1

X1 - Real array of ratio of reflectances calculated from 1 and 2

D1 - Real array of the difference between X1 and Y1 elements

E - Real array (printer output) of 1, 0, and -1 calculated (using tests on D1 and T1)

E1 - Real array (alphanumeric printer output) of 1, 0, and -1 calculated (using tests on E)

TBB - Real array of calculated brightness temperature

R4 - Real array of reflectances calculated from channel 4

LC - Number of columns recorded for channel 1

LR - Number of rows recorded for channel 1

LC2 - Number of columns recorded for channel 2

LR2 - Number of rows recorded for channel 2

LR3 - Number of rows recorded for channel 3

LC3 - Number of columns recorded for channel 3

LR4 - Number of rows recorded for channel 4

LC4 - Number of columns recorded for channel 4

Plot 1 - Logical indicator of whether temperature (of channel 3) plot requested

Plot 2 - Logical indicator of whether reflectance (of channel 1) plot requested

Plot 3 - Logical indicator of whether cloud optical thickness (channels 1 and 2) plot requested

Plot 4 - Logical indicator of whether delts ratio [plus liquid minus ice] (channels 1 and 2) plot requested

Plot 5 - Logical indicator of whether delta ratio [plus liquid minus ice] with H2O vapor (channels 2 and 4) plot requested

LR, LR2, LR3, and LR4 $\leq$ 67, while LC, LC2, LC3, and LC4 $\leq$ 35.

Input and Output:

Unit 5 - NAMELIST INPUT

NCHAN1 - Actual number name of first channel requested (to be used for reflectance ratio, and delta calculation)

NCHAN2 - Actual number name of second channel requested (to be used for reflectance ratio, and delta calculation)

NUMPIT - Skip factor for pixel access

NUMLIN - Skip factor for line access

IPIXEL - First pixel on line which user desires plotted (in current version, this is overridden)

NCHAN3 - Actual number name of third channel requested to be used for temperature calculations

NCHAN4 - Actual number name of fourth channel requested to be used for reflectance and delta ratio calculations

Plot 1 - Logical variable indicating whether temperature (channel 3) plot is desired

Plot 2 - Logical variable indicating whether reflectance (channel 1) plot is requested

Plot 3 - Logical variable indicating whether cloud optical thickness (channels 1 and 2) plot is requested

Plot 4 - Logical variable indicating whether delta ratio [plus liquid minus ice] (channels 1 and 2) is requested

Plot 5 - Logical variable indicating whether delta ratio [plus liquid minus ice] (channels 2 and 4) with H2O vapor is requested

NUMPLT - Number of plots requested

Unit 6 - Printer unit listing program status messages and calculated values

Error Returns: Since channels 15 and 16 are not active, a request for calculations and/or plots involving these channels results in an explanatory error message being printed and termination of the program. Since there are only 22 channels possible, a request for a channel number greater than 22 results in an explanatory error message being printed and in termination of the program.

Called Subroutines: RD192, TEMBRH, REFLE1, NOSTAE (system routine), PLOTST (Wolf routine)

4.1.1.5 Program or Subroutines To Use MAIN

N/A

4.1.1.6 References

N/A

4.1.2 TEMBRH

4.1.2.1 Identification

English Language Title: S192 brightness temperature calculation and plotting routine

Programming Language: FORTRAN

Analyst's Name and Organization: F. Milasi, Computer Sciences Corporation

Programmer's Name and Organization: F. Milasi, Computer Sciences Corporation

4.1.2.2 Functional Procedure

TEMBRH calculates the equivalent brightness temperature of the input channel's data and calls routines which display its results.

4.1.2.3 Formal Procedure

The average wavelength for the channel is selected from the wavelengths possible by checking the channel number requested. This is then used in the inverse Planck function to calculate brightness temperature. The formula used:

$$TBB = 1.4388E + 04/\{WAVET * LOG\ [1. + (3.7413E + 04/(\pi * WAVET^{5} * RAD))]\}$$

where TBB is the equivalent brightness temperature

WAVET is the determined associated wavelength

RAD is the radiance value for the pixel

The listing of the results is made using subroutine WRITT and, if PLOT1 value indicates the desirability of it, the plot is made through subroutine CONTOU. Control then returns to the call program.

4.1.2.4 Coding Information

Calling Sequence: CALL TEMBRH (NCHAN3, C, NUMPLT)

where NCHAN3 is the number-name of the channel for which a temperature plot is requested

C is the real array containing the selected subset of radiance values associated with the NCHAN3 wavelength band

NUMPLT is the number of plots requested

COMMON Storage Used:

VALUES: Refer to Section 4.1.1.4.

Input and Output:

Unit 6 - Printer unit for error message

Error Returns: If the requested wavelength channel number name for NCHAN3 does not lie within the number 1 through 21, an explanatory message is printed, no further calculation is performed, and the program terminates.

Called Subroutines: WRITT, CONTOU

4.1.2.5 Programs or Subroutines To Use TEMBRH

MAIN (of QUICK)

4.1.2.6 References

N/A

4.1.3 REFLE1

4.1.3.1 Identification

English Language Title: S192 Reflectance calculation and plotting routine for three channels

Programming Language: FORTRAN

Analyst's Name and Organization: F. Milasi, Computer Sciences Corporation

Programmer's Name and Organization: F. Milasi, Computer Sciences Corporation

4.1.3.2 Functional Procedure

REFLE1 calculates the reflectance of the first input channel's data (i.e., data from NCHAN1) and calls routines which list and/or display its results.

4.1.3.3 Formal Procedure

The irradiance value for the channel is selected from the corresponding irradiance values possible by checking the channel number requested. The

array, FO, from which it is selected was secured from interpolation of irradiance values supplied by Dr. Thekakara (GSFC) using the average wavelength of the various channels possible for the interpolation independent variable input. The cosine of the input zenith angle is multiplied by this irradiance value and then the product is divided into π. This factor is multiplied by each input radiance to yield reflectance. The applicable formula is

$$R1 = A * \pi/(\cos(Z) * IRR)$$

where R1 is the calculated reflectance

A is the radiance value

Z is the zenith angle

IRR is the associated irradiance value

The calculated value R1 is then multiplied by 1000 to enable fixed point representation of the data when plotted. Data is listed through a call to subroutine WRITT. If a reflectance plot is desired (indicated by the value of PLOT2), the contour is plotted through subroutine CONTOU.

If only one channel was specified, an appropriate message is printed and the program terminates. Otherwise, optical thickness is calculated through subroutine CLUTHK. Reflectance of channel 2 (a channel in which ice and water differ in their optical properties) and the ratio of that reflectance and the one calculated in this routine (from a channel in which ice and water are the same in their optical properties) will be calculated through the call to REFLE2. Control then returns to the calling program.

4.1.3.4 Coding Information

Calling Sequence: CALL REFLE1 (A, ANG, NCHAN1, B, ANG2, NCHAN2, D, ANG4, NCHAN4, NUMPLT)

where A is the subset real array of radiance data associated with channel 1

ANG is the current zenith angle associated with channel 1 data

NCHAN1 is the number name of the first channel requested

B is the subset real array of radiance data associated with channel 2

ANG2 is the current zenith angle associated with channel 2 data

NCHAN2 is the number name of channel 2 requested

D is the subset real array of radiance data associated with channel 4

ANG4 is the current zenith angle associated with channel 4 data

NCHAN4 is the number name of channel 4 requested

NUMPLT is the number of plots requested

COMMON Storage Used:

VALUES: Refer to Section 4.1.1.4

Input and Output:

Unit 6 - Printer unit for error messages

Error Returns: If the requested wavelength channel number name for NCHAN1 does not lie within the numbers 1 through 22, an explanatory message is printed, no further calculation is performed, and the program terminates.

Called Subroutines: WRITT, CONTOU, CLUTHR, REFLE2

4.1.3.5 Programs or Subroutines To Use REFLE1

MAIN (of QUICK)

4.1.3.6 References

N/A

4.1.4 REFLE2

4.1.4.1 Identification

English Language Title: S192 reflectance calculation and plotting routine for two channels

Programming Language: FORTRAN

Analyst's Name and Organization: F. Milasi, Computer Sciences Corporation

Programmer's Name and Organization: F. Milasi, Computer Sciences Corporation

4.1.4.2 Functional Procedure

REFLE2 calculates the reflectance of the first input channel's data (i.e., data from NCHAN2) and the ratio of the reflectances of NCHAN2 data to NCHAN1 data, and calls routines which list and/or display its results.

4.1.4.3 Formal Procedure

As in REFLE1, the irradiance value for the channel is selected from the corresponding irradiance values possible by checking the channel number requested for NCHAN2. The array, FO, from which it is selected, is the same array used in REFLE1, (i.e., that which was secured from interpolation of irradiance values supplied by Dr. Thekakara of GSFC using the average wavelength of the various channels possible for the interpolation independent variable input). The cosine of the input zenith angle is multiplied by this irradiance value and the product is divided into π. This factor is multiplied by each input radiance to yield reflectance. The applicable formula is

$$R2 = B * \pi/(\cos(Z) * IRR)$$

where R2 is the calculated reflectance

B is the radiance value

Z is the zenith angle

IRR is the associated irradiance value

If a value of NCHAN4 is present, subroutine REFLE3 is called to yield results similar to REFLE2 for channels 2 and 4. In any case, the ratios of the reflectances calculated from NCHAN2 to those calculated from NCHAN1 are calculated. This yields the ratio of a channel in which the optical properties of ice and water differ to one in which the optical properties of ice and water are similar. These ratio values are multiplied by 100 for listing purposes in subroutine WRITT.

The equivalent ratio values which correspond to the optical thickness values calculated using NCHAN1, are obtained through subroutine CLUTHK. The difference in R2:R1 ratio values is calculated. This difference reveals the thermodynamic phase of the cloud particles (a minus value is ice, a plus value is water, and zero is undetermined). The difference array is multiplied by 100, stored in array RR, and listed using subroutine WRITT.

If any NCHAN1 optical thickness is less than or equal to 8, the corresponding value in the array RR is set to zero (for clarity in its display by CONTOU). If requested by the user through logical indicator PLOT4, a contour of the resultant RR array is made. Finally, the water-ice numeral and alphanumeric plots are made using subroutine DELTA1. Control then returns to the calling program.

4.1.4.4 Coding Information

Calling Sequence: CALL REFLE2 (B, ANG2, NCHAN2, FF, D, ANG4, NCHAN4, NUMPLT)

where B is the array of radiance values associated with NCHAN2 (i.e., the channel it represents)

ANG2 is the zenith angle associated with NCHAN2

NCHAN2 is the number name of the requested scan channel

FF is the array of irradiance values associated with each of the average wavelengths of all of the channels

D is the array of radiance values associated with NCHAN4 (i.e., the channel it represents)

ANG4 is the zenith angle associated with NCHAN4

NCHAN4 is the number name of the requested scan channel

NUMPLT is the number of plots requested

COMMON Storage Used:

VALUES: Refer to Section 4.1.1.4

Input and Output:

Unit 6 - Printer unit for error messages

Error Returns: Refer to Section 4.1.3.4

Called Subroutines: REFLE3, WRITT, CLUTHK, CONTOU, DELTA1

4.1.4.5 Programs or Subroutines To Use REFLE2

REFLE1

4.1.4.6 References

N/A

4.1.5 REFLE3

4.1.5.1 Identification

English Language Title: S192 reflectance calculation and plotting routine for one channel

Programming Language: FORTRAN

Analyst's Name and Organization: F. Milasi, Computer Sciences Corporation

Programmer's Name and Organization: F. Milasi, Computer Sciences Corporation

4.1.5.2 Functional Procedure

REFLE3 calculates the reflectance of the fourth input channel's data (i.e., data from NCHAN4) and the ratio of reflectances of NCHAN2 data to NCHAN4 data, and calls routines which list and/or display its results.

4.1.5.3 Formal Procedure

As in REFLE1 and REFLE2, the irradiance data pertinent to the channel requested is selected from the stored irradiance data for all channels by checking the channel number requested for NCHAN4. The cosine of the input zenith angle is multiplied by the selected irradiance value and then that product is divided into π. This quotient is multiplied by each input radiance to yield reflectance. The applicable formula is

$$R4 = D * \pi/(\cos(Z) * IRR)$$

where R4 is the calculated reflectance

D is the radiance value

Z is the zenith angle

IRR is the associated irradiance value

The ratio of reflectances calculated for NCHAN2 and NCHAN4 is then calculated. As in REFLE2, this is the ratio of a channel in which the optical characteristics of water and ice differ to one in which the optical characteristics of water and ice are similar. The resulting matrix of values is multiplied by 100 for listing purposes in subroutine WRITT (which is then called).

The equivalent R2:R4 ratio values which correspond to the optical thickness values calculated using NCHAN1 are obtained through subroutine CLUTHK. The difference in R2:R4 ratio values is calculated, multiplied by 100, stored in array RR, and listed using subroutine WRITT. Again, this difference array represents the thermodynamic phase of cloud particles (where a negative result means ice, a positive result means water, and zero is undetermined).

If any NCHAN1 optical thickness is less than or equal to 8, the corresponding value in the array RR is set to zero (for clarity in its display by CONTOU). If requested by the user through logical indicator PLOT5, a contour of the resultant RR array is made. Finally, the water-ice numeral and alphanumeric plots are made using subroutine DELTA1. Control returns to the calling program.

4.1.5.4 Coding Information

Calling Sequence: CALL REFLE3 (FF, ANG4, NCHAN4, D, R2, NUMPLT)

where
- FF is the array of irradiance values associated with each of the average wavelengths of all of the channels
- ANG4 is the zenith angle associated with NCHAN4
- NCHAN4 is the number name of the requested scan channel
- D is the real array of radiance values associated with NCHAN4 (i.e., the channel it represents)
- R2 is the real array of calculated reflectance associated with NCHAN2 (i.e., the channel it represents)
- NUMPLT is the number of plots requested

COMMON Storage Used:

VALUES: Refer to Section 4.1.1.4

Input and Output:

Unit 6 - Printer unit for error messages

Error Returns: Refer to Section 4.1.3.4

Called Subroutines: WRITT, CLUTHK, CONTOU, DELTA1

4.1.5.5 Programs or Subroutines To Use REFLE3

REFLE2

4.1.5.6 References

N/A

4.1.6 CLUTHK

4.1.6.1 Identification

English Language Title: S192 equivalent optical thickness - ratio determinating routine

Programming Language: FORTRAN

Analyst's Name and Organization: F. Milasi, Computer Sciences Corporation

Programmer's Name and Organization: F. Milasi, Computer Sciences Corporation

4.1.6.2 Functional Procedure

Depending upon the input request, equivalent optical thickness is calculated and its plot made, or equivalent channel 2:channel 1 reflectance ratio or channel 2: channel 4 reflectance ratio is calculated.

4.1.6.3 Formal Procedure

If the optical thickness values are desired as indicated by input value NUM, the SPLINE interpolation routine is used to determine them using the reflectance of channel 1 as the input independent variable.

The second derivative at the upper and lower bands is set to zero. This is based on the nature of the data used for interpolation. The quadrant 1 confined graph of reflectance versus optical thichness is nearly linear (for small values of reflectance and optical thickness and asymtomatically approaches the REFLEC = 1 vertical line for large values of optical thickness. In either case,

the second derivative is zero or nearly zero and (for practical purposes) is set to zero. The resulting array is listed using subroutine WRITT.

If indicated by the value of the input indicator PLOT3, the calculated array is contoured through subroutine CONTOU. Control would then return to the calling program.

If the equivalent channel 2:channel 1 reflectance ratio is desired as indicated by the input variable NUM, the SPLINE interpolation routine is used to determine this ratio using the optical thickness calculated from channel 1 as the input independent variable.

The upper bound of the first derivative and the lower bound of the second derivative are set to zero. This is based on the nature of the interpolation data graph. The quadrant 1 confined optical thickness versus ratio graph is nearly a straight line for small values of optical thickness (so its second derivative there is zero), and is nearly a horizontal line for large values of optical thickness (so its first derivative there is zero). Control would then return to the calling program.

If the equivalent channel 2:channel 4 reflectance ratio is desired as indicated by the input variable NUM, the SPLINE interpolation routine is used to determine this ratio using the optical thickness calculated from channel 1 as the input independent variable.

The upper bound of the first derivative and the lower bound of the second derivative are set to zero. This is based on the nature of the interpolation data graph (similar to the one present for channel 2:channel 1 ratio/optical thickness). Control would then return to the calling program.

4.1.6.4 Coding Information

Calling Sequence: CALL CLUTHK (NUM, NUMPLT)

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

where NUM is an indicator of output desired:
= 0, optical thickness equivalent of the radiances of channel NCHAN1 is desired
= 1, ratio (NCHAN2:NCHAN1) equivalent of the optical thickness calculated from NCHAN1 is desired
= 2, ratio (NCHAN2:NCHAN4) equivalent of the optical thickness calculated from NCHAN1 is desired

NUMPLT is the number of plots desired

COMMON Storage Used:

VALUES: Refer to Section 4.1.1.4

Input and Output: N/A

Error Returns: N/A

Called Subroutines: SPLINE, WRITT, CONTOU

4.1.6.5 Programs or Subroutines To Use CLUTHK

REFLE1, REFLE2, REFLE3

4.1.6.6 References

N/A

4.1.7 DELTA1

4.1.7.1 Identification

English Language Title: Thermodynamic phase of cloud determination program

Programming Language: FORTRAN

Analyst's Name and Organization: F. Milasi, Computer Sciences Corporation

Programmer's Name and Organization: F. Milasi, Computer Sciences Corporation

4.1.7.2 Functional Procedure

DELTA1 generates numeric and alphanumeric arrays of data which indicate the thermodynamic phase of the cloud particles and calls a routine to list these arrays.

4.1.7.3 Formal Procedure

If the value of the input variable NUMM indicates that the difference between the actual ratio of the second channel's reflectance (NCHAN2, in which ice and water differ in their optical characteristics) to the first channel's reflectance (NCHAN1, in which ice and water have similar optical characteristics), and the calculated ratio, is to be analyzed (i.e., NUMM is zero), processing begins using the appropriate limits LC2 and LR2.

If the difference in ratios is negative and if the optical thickness of the cloud is greater than 8, ice is present, so the corresponding indicators, -1 for numeric array and an I for the alphanumeric array, are stored in that pixel position.

If the difference in ratios is zero or if the optical thickness of the cloud is less than 8, no conclusion can be drawn, so the corresponding indicators, 0 for the numeric array and blanks for the alphanumeric array, are stored in that pixel position.

Finally, if the difference in ratios is positive and if the optical thickness of the cloud is greater than 8, water is present, so the corresponding indicators, +1 for the numeric array and zeros for the alphanumeric array, are stored in that pixel position.

Subroutine WRITT is next called to list these generated numeric and alphanumeric arrays, and control returns to the calling program.

If the value of the input variable NUMM was 1, the difference between the actual ratio of the second channel's reflectance (NCHAN2, in which ice and water differ in their optical characteristics) to the fourth channel's reflectance (NCHAN4, in which ice and water have similar optical characteristics), and the calculated ratio, is to be analyzed. Processing identical to that for a value of NUMM=0 begins using appropriate limits of LC4 and LR4. The listing of the generated numeric and alphanumeric arrays is again made through subroutine WRITT, and control returns to the calling program.

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

4.1.7.4 Coding Information

Calling Sequence: CALL DELTA1 (NUMM)

where NUMM is an indicator of desired processing:
= 0, calculations involve NCHAN2:NCHAN1 ratios
= 1, calculations involve NCHAN2:NCHAN4 ratios

COMMON Storage Used:

VALUES: Refer to Section 4.1.1.4

Input and Output: N/A

Error Returns: N/A

Called Subroutines: WRITT

4.1.7.5 Programs or Subroutines To Use DELTA1

REFLE2, REFLE3

4.1.7.6 References

N/A

4.1.8 CONTOU

4.1.8.1 Identification

English Language Title: S192 Wolf contour generation routine

Programming Language: FORTRAN

Analyst's Name and Organization: F. Milasi, Computer Sciences Corporation

Programmer's Name and Organization: F. Milasi, Computer Sciences Corporation

4.1.8.2 Functional Procedure

CONTOU utilizes Wolf Plotting routines to contour input S192 data.

4.1.8.3 Formal Procedure

Minima and maxima values for indices for contours are set using input variables. Frame limits of plot and letter sizes are set using WOLFPLOT routines SETGRD and SETSIZ. Then, depending upon the input variable J, any one of five plots is initialized and a title inserted through a call to HORLIN. The five possibilities are: temperature brightness, reflectances, cloud optical thickness, delta ratio (plus liquid minus ice), and delta ratio (plus liquid minus ice) with H2O vapor - NCHAN2:NCHAN4.

For all plots, the print size is halved for the plot itself through a new call to SETSIZ. The format desired for the value labels on the contours is set through CONFRM. Subroutine INDEX requires all lines to be solid (no spacing). The subroutine LABELI call eliminates the line underneath any label (i.e., contour line goes to one side of label, stops, and starts again on the other side).

The subroutine OGRID call sets the limits of the plot. The call to subroutine ZDUMP is used for debugging, listing the array so that it corresponds to the X and Y directions of the contour plot.

Subroutine UNDEF defines all zero values as "undefined" values, while FILL requests that all such undefined values be replaced by the average of all "neighboring" (apropos of the contour configuration) defined values.

The actual contour plot is created through a call to TRACER and the FRAME is advanced, through FRMADV, in preparation for subsequent plots. If this is the last plot (verified by checking a program access counter against the input number of plots desired, NUMPLT), the frame counter is set to 999 and plot output device use closed through SETBLK and ENDPLT.

4.1.8.4 Coding Information

Calling Sequence: CALL CONTOU (TTT, LRW, LCW, NUMPLT, J)

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

where TTT is the array to be contoured

LRW is the array row index limit

LCW is the array column index limit

NUMPLT is the total number of plots to be made in the entire run

J is the indicator of plot type desired:
= 1, brightness temperature
= 2, reflectances
= 3, cloud optical thickness
= 4, delta ratio (+ liquid-ice)
= 5, delta ratio (+ liquid-ice) with H2O vapor

COMMON Storage Used:

OPTION: IMIN - Minimum index for row

IMAX - Maximum index for row

JMIN - Minimum index for column

JMAX - Maximum index for column

FINTER - Specifies contour interval if NUM=0

These specify the subarray of the data array to be contoured.

NUMV - Number of specific contours if FINTER=0

CONTUR - Array of specific contours if NUMV≠0

This COMMON is required for the Wolf contour package.

Input and Output: Plottape for Wolf contour output

Error Returns: N/A

Called Subroutines: SETGRD, SETSIZ, HORLIN, CONFRM, INDEX, LABELI, OGRID, ZDUMP, UNDEF, FILL, TRACER, FRMADV, SETBLK, ENDPLT (all Wolf contour package routines)

4.1.8.5 Programs or Subroutines To Use CONTOU

TEMBRH, REFLE1, REFLE2, REFLE3, CLUTHK

4.1.8.6 References

GSFC Computer Program Library, A00227, Wolf Plotting and Contouring Package - (Rev. 11-72), G. T. Masaki (CSC), January 1972 (revised in November 1972)

4.1.9 WRITT

4.1.9.1 Identification

English Language Title: S192 calculated array listing program

Programming Language: FORTRAN

Analyst's Name and Organization: F. Milasi, Computer Sciences Corporation

Programmer's Name and Organization: F. Milasi, Computer Sciences Corporation

4.1.9.2 Functional Procedure

WRITT lists the contents of arrays calculated in other parts of the QUICK package as indicated through the call argument requests.

4.1.9.3 Formal Procedure

The sole call argument, J, merely indicates the array to be plotted and its content. Depending upon that value, any one of nine possible label and array list logical sequences are followed. Within the sequences, the listing loop limits correspond to the limits of the generating array (i.e., LR, LR2, LR3, or LR4 for row limit; LC, LC2, LC3, or LC4 for column limit). The output format also corresponds to data type (i.e., F4.0 for optical thicknesses, reflectances, reflectance ratios, delta ratios, and brightness temperatures; A4 for delta ratio symbol maps).

4.1.9.4 Coding Information

Calling Sequence: CALL WRITT (J)

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

where J is an indicator of listing desired
= 1, lists reflectances
= 2, lists cloud optical thickness
= 3, lists reflectances ratio R(1.61 μm)/R(0.74 μm)
= 4, lists delta ratio (+ liquid-ice) NCHAN2/NCHAN1 or R(1.61 μm)/R(0.74 μm)
= 5, lists delta ratio (+ liquid-ice) NCHAN2/NCHAN1 or R(1.61 μm)/R(0.74 μm)
= 6, lists temperature brightness
= 7, lists reflectance ratio R(1.61 μm)/R(1.14 μm)
= 8, lists delta ratio (+ liquid-ice) with H2O vapor NCHAN2/NCHAN4 or R(1.61 μm)/R(1.14 μm)
= 9, lists the delta ratio (+ liquid-ice) with H2O vapor symbol map NCHAN2/NCHAN4 or R(1.61 μm)/R(1.14 μm)

COMMON Storage Used:

VALUES: Refer to Section 4.1.1.4

Input and Output:

Unit 9 - Printer unit containing the listing of the data (necessitates use of corresponding JCL; all parameters similar to JCL for default unit 6 except DD name is changed)

Error Returns: N/A

Called Subroutines: N/A

4.1.9.5 Programs or Subroutines To Use WRITT

TEMBRH, REFLE1, REFLE2, REFLE3, CLUTHK, DELTA1

4.1.9.6 References

N/A

4.1.10 SPLINE

4.1.10.1 Identification

English Language Title: SPLINE interpolation routine

Programming Language: FORTRAN

Analysts' Names and Organization: J. L. Walsh, J. H. Ahlberg, and E. N. Nilson, GSFC

Programmers' Names and Organization: I. Harris and S. Selzer, GSFC

4.1.10.2 Functional Procedure

This routine interpolates among input data using the SPLINE method of interpolation.

4.1.10.3 Formal Procedure

See Section 4.1.10.6.

4.1.10.4 Coding Information

Calling Sequence: CALL SPLINE (S, X, Y, N, IN, T, IL, VL, IU, VU, E, U)

where S is the real input independent variable for which an interpolated dependent variable is needed

X is the real array of independent variables of input data

Y is the real array of dependent variables of input data

N is the number of elements in X (and in Y). It is their dimension

IN is the indicator of whether initialization or actual interpolation is requested
= 1, initialization
= 2, interpolation

T is the real output calculated dependent variable corresponding to S derived through interpolation

IL and IU are as indicators
= 1, parabolic runout conditions at lower and upper bound
= 2, first derivative (VL, VU) at lower or upper bound
= 3, second derivative (VL, VU) at lower or upper bound

VL lower and VU upper are bound values of first or second derivative as specified by IL and IU

E and U are work area arrays needed by program whose dimension is greater than or equal to N

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

COMMON Storage Used: N/A

Input and Output: N/A

Error Returns: N/A

Called Subroutines: N/A

4.1.10.5 Programs or Subroutines To Use SPLINE

CLUTHK (QUICK version), CLUTHK (QUICK3 version)

4.1.10.6 References

The Theory of Splines and Their Application. J. S. Walsh, J. H. Ahlberg, and E. N. Nilson: Academic Press, 1967

4.1.11 Hardware Requirements

One OS/360 printer for output listings and error messages
Two OS/360 tape drives for input tape and WOLFPLOT tape
One OS/360 card reader for namelist input

4.1.12 Core Requirements

The core requirement for the entire program (including required Wolf package) is 452K bytes.

4.1.13 Time Requirements

The number of plots requested will affect this value; normally, an average run takes 5 minutes CPU and 4 minutes IO.

4.1.14 Card Input

See Appendix H.

4.1.15 Data Card Format

See listing in Appendix H.

4.1.16 Output

See Appendix H.

4.2 QUICK3

4.2.1 MAIN

4.2.1.1 Identification

English Language Title: S192 general plot driver program

Programming Language: FORTRAN

Analyst's Name and Organization: F. Milasi, Computer Sciences Corporation

Programmer's Name and Organization: F. Milasi, Computer Sciences Corporation

4.2.1.2 Functional Procedure

This program, through subroutine calls and a minimal amount of calculation, accesses S192 data tapes and plots as 200-by-200 subset of the scene. An entire scene can be formed through the assembling of such overlapping subsets.

4.2.1.3 Formal Procedure

Through NAMELIST INPUT, user requests for the channel to be printed, the output unit to be used, starting line, starting pixel, line skip factor, pixel skip factor, and desired calculation and plot are recorded. The channel is read by the actual channel number recorded by JSC, not by its order on the tape (since many are missing). Since channels 15 and 16 had been missing from all S192 tapes received, a check is made that the user did not erroneously request one of these. This program, unlike QUICK, was used with the expected line straightened data and was not modified to accommodate the several erroneously sent conical scan data tapes.

In preparation for the access logic, which only accepts data at or beyond the first line number and at every line skip factor beyond it, the value of the first

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

line to be accepted is checked against skip factor. If the first line number is larger than the skip factor, the value by which the data line number must be decremented before evaluating it modulo the line skip factor for determining whether it is to be stored, is changed to ensure that the first line desired is also stored.

The S192 data tape is accessed via the call to subroutine RD192. If the data lies beyond the first requested line and covers the channel requested, it is tested to determine whether modulo the line skip factor is desired.

The line (i.e., that portion of it specified by starting pixel number and skip factor) is then stored. The zenith angle is inverted to degrees and fractions of a degree and stored. If either all 200-by-200 pixels have been acquired or an end of file is encountered, then the requested plot is made through a call to an appropriate subroutine (TEMBRH for brightness temperature; REFLE1 for reflectance or optical thickness) is made. In either case, the program then terminates.

4.2.1.4 Coding Information

Calling Sequence: N/A

COMMON Storage Used:

VALUE: LR3 - Number of rows recorded (LR3 ≤ 200)

LC3 - Number of columns recorded (LC3 ≤ 200)

TBB - Real array of recorded radiance values for selected pixels

PLTTEM - Logical indicator of desire for temperature calculation and plot

PLTREF - Logical indicator of desire for reflectance calculation and plot

PLTCLD - Logical indicator of desire for cloud optical thickness calculation and plot

ANG - Zenith angle associated with scene

Input and Output:

Unit 6 - Printer unit for status and error messages

Unit 5 - Card reader for input NAMELIST INPUT:

NCHAN1 - Desired number name of data channel

NUMPIT - Pixel skip factor

NUMLIN - Line skip factor

IPIXEL - Starting pixel number

ISTLIN - Starting line number

PLTTEM - Logical indicator of desire for temperature calculation and plot

PLTREF - Logical indicator of desire for reflectance calculation and plot

PLTCLD - Logical indicator of desire for cloud optical thickness calculation and plot

Error Returns: If an error is encountered in reading the tape, an appropriate message is printed and the program terminates. If the user requests channels 15 or 16 (which are not present on any S192 tapes received from JSC), an appropriate explanatory error message is printed and the program terminates.

Called Subroutines: RD192, TEMBRH, REFLE1

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

4.2.1.5 Programs or Subroutines To Use MAIN

N/A

4.2.1.6 References

N/A

4.2.2 TEMBRH (Version 2)

4.2.2.1 Identification

English Language Title: S192 brightness temperature calculation and plotting routine (version 2)

Programming Language: FORTRAN

Analyst's Name and Organization: F. Milasi, Computer Sciences Corporation

Programmer's Name and Organization: F. Milasi, Computer Sciences Corporation

4.2.2.2 Functional Procedure

TEMBRH (version 2), like TEMBRH (version 1), calculates the equivalent brightness temperature of the input channel's data and calls routines which display its results.

4.2.2.3 Formal Procedure

The average wavelength for the channel is selected from the wavelengths possible by checking the channel number requested. This is then used in the inverse Planck function to calculate brightness temperature. The formula used is

$$TBB = 1.4388E04/\{WAVET * LOG\,[1. + (3.7413E04/(\pi * WAVET^{5} * RAD))]\}$$

where TBB is the desired brightness temperature

WAVET is the determined associated average wavelength

RAD is the radiance value for the pixel

Subroutine CONTOU (version 2) is then called to plot the temperatures calculated. Finally, control returns to the calling program.

4.2.2.4 Coding Information

Calling Sequence: CALL TEMBRH (NCHAN1)

where NCHAN1 is the number name of the desired channel

COMMON Storage Used:

VALUE: Refer to Section 4.2.1.4

Input and Output: N/A

Error Returns: N/A

Called Subroutines: CONTOU

4.2.2.5 Programs or Subroutines To Use TEMBRH

MAIN (of QUICK3)

4.2.2.6 References

N/A

4.2.3 REFLE1 (Version 2)

4.2.3.1 Identification

English Language Title: S192 reflectance calculation and plotting routine for a single channel

Programming Language: FORTRAN

Analyst's Name and Organization: F. Milasi, Computer Sciences Corporation

Programmer's Name and Organization: F. Milasi, Computer Sciences Corporation

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

4.2.3.2 Functional Procedure

REFLE1 calculates the reflectance of the input channel, plots reflectance through a subroutine call, and calculates and plots cloud optical thickness through a subroutine call.

4.2.3.3 Formal Procedure

The irradiance value for the channel is selected from the corresponding irradiance values possible by checking the channel number requested. The array FF from which it is selected was secured through interpolation of irradiance values supplied by Dr. Thekakara (GSFC) using the average wavelength for the various channels possible for the interpolation independent variable.

The cosine of the zenith angle wich was passed through COMMON) is multiplied by this irradiance value and that product is then divided into π . The factor produced is utilized repeatedly in the reflectance calculation loop. The applicable formula is

$$R1 = TBB * \pi/(\cos(Z) * IRR)$$

where R1 is the calculated reflectance

TBB is the input radiance value for the pixel

Z is the zenith angle

IRR is the appropriate corresponding irradiance value

If optical thickness calculation and plot is desired, subroutine CLUTHK is called. The reflectance array, multiplied by 1000, is plotted by a call to subroutine CONTOU. The program then terminates.

4.2.3.4 Coding Information

Calling Sequence: CALL REFLE1 (NCHAN1)

where NCHAN1 is the input channel number name

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

COMMON Storage Used:

VALUE: Refer to Section 4.2.1.4

Input and Output:

Unit 6 - Printer unit for error message

Error Returns: If the requested channel number does not lie between numbers 1 and 22, an appropriate error message is printed and the program terminates.

Called Subroutines: CLUTHK, CONTOU

4.2.3.5 Programs or Subroutines To Use REFLE1

MAIN (of QUICK3)

4.2.3.6 References

N/A

4.2.4 CONTOU (Version 2)

4.2.4.1 Identification

English Language Title: S192 Wolf contour generation routine (version 2)

Programming Language: FORTRAN

Analyst's Name and Organization: F. Milasi, Computer Sciences Corporation

Programmer's Name and Organization: F. Milasi, Computer Sciences Corporation

4.2.4.2 Functional Procedure

CONTOU (version 2) utilizes Wolf plotting routines to contour input arrays calculated from S192 data

4.2.4.3 Formal Procedure

Maxima and minima values for indices for contours are set using variables input from COMMON. Plotting development is begun through calls to subroutines

NOSTAE, PLOTST, and SETSIZ. Depending upon the logical values of the plot indicators, any one of three plots is initialized and a title printed through a call to HORLIN. The three possibilities are brightness temperature, reflectance, or cloud optical thickness. The frame size is then set through subroutine SETGRD, the letter size changed through subroutine SETSIZ, and the format of the contour labels set through subroutine CONFRM. Subroutine INDEX requires all lines to be solid (no spacing), while subroutine LABELI eliminates the contour line underneath the label for that line.

The call to subroutine OGRID sets the limits of the plot while the call to UNDEF defines all zero entries as undefined. Subroutine FILL replaces all undefined values with the average of all neighboring defined values. The actual plot is created through a call to subroutine TRACER and the frame is advanced through FRMADV. Since only one plot is to be made, the frame counter is set to 999 and recorded, and the plotting is completed through calls to SETBLK and ENDPLT. Control then returns to the calling program.

4.2.4.4 Coding Information

Calling Sequence: CALL CONTOU (no arguments)

COMMON Storage Used:

VALUE--Refer to Section 4.2.1.4

OPTION--Refer to Section 4.1.8.4

BUF$--BUFFER: storage buffer needed for large contours by Wolf contour program

Both OPTION and (because of plot size desired) BUF$ are required for the utilization of the Wolf contour package.

Input and Output: Plottape for Wolf contour output

Error Returns: N/A

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

Called Subroutines: NOSTAE (OS/360 system routine), PLOTST, SETSIZ, HORLIN, SETGRD, CONFRM, INDEX, LABEL1, OGRID, UNDEF, FILL, TRACER, FRMADV, SETBLK, ENDPLT (all Wolf contour package routines)

4.2.4.5 Programs or Subroutines To Use CONTOU

TEMBRI1 (version 2), REFLE1 (version 2), CLUTHK (version 2)

4.2.4.6 References

N/A

4.2.5 CLUTHK (Version 2)

4.2.5.1 Identification

English Language Title: S192 equivalent optical thickness determining routine (version 2)

Programming Language: FORTRAN

Analyst's Name and Organization: F. Milasi, Computer Sciences Corporation

Programmer's Name and Organization: F. Milasi, Computer Sciences Corporation

4.2.5.2 Functional Procedure

CLUTHK (version 2) calculates equivalent optical thickness and generates a plot through a subroutine call.

4.2.5.3 Formal Procedure

The SPLINE interpolation routine is used to determine equivalent optical thickness using the previously calculated reflectance of the channel. The first call to SPLINE initializes the frequently utilized variables; the second call (in the loop) actually does the interpolation.

The second derivative at both the upper and lower limits is set to zero because of the nature of the data which is being interpolated. The quadrant 1 confined

graph of reflectance versus optical thickness is nearly linear for small values of optical thickness and asymptotically approaches the REFLEC=1 vertical line for extremely large values of optical thickness. In either case, the second derivative is nearly zero (and for all practical purposes acts as such).

After calculation, subroutine CONTOU is used to plot the derived optical thickness. Control then returns to the calling program.

4.2.5.4 Coding Information

Calling Sequence: CALL CLUTHK (no arguments)

COMMON Storage Used:

VALUE: Refer to Section 4.2.1.4

Input and Output: N/A

Error Returns: N/A

Called Subroutines: SPLINE, CONTOU

4.2.5.5 Programs or Subroutines To Use CLUTHK

REFLE1 (version 2)

4.2.5.6 References

N/A

4.2.6 Hardware Requirements

One OS/360 printer for output listings and error messages
Two OS/360 tape drives for input tape and WOLFPLOT tape
One OS/360 card reader for NAMELIST INPUT

4.2.7 Core Requirements

The core requirement for the entire program (including the required Wolf contour package) in 452K bytes.

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

4.2.8 Time Requirements

An average run takes 5 minutes CPU and 4 minutes IO.

4.2.9 Card Input

See Appendix I.

4.2.10 Data Card Format

See Appendix I.

4.2.11 Output

See Appendix I.

APPENDIX A - LIST191 RUN PROGRAM LISTINGS, INPUT, AND JCL OUTPUT

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
///RMJM191 JOB (S20272604D,T,L00360,001H01),JDB,MSGLEVEL=1
//* THISDATE READ AND LIST S-191 TAPES
// EXEC FORTRANH,PARM='OPT=2,EBCDIC,LIST,ID,MAP,XREF',REGION=235K
//SOURCE.SYSIN DD *
C
C
C       THIS PROGRAM WAS WRITTEN BY M.MACK OF C.S.C. FOR TASK 092(SKYLAB)
C       UNDER CONTRACT NAS-5 - 11999 IN SEPTEMBER-OCTOBER,1973. IT
C       UTILIZES SUBROUTINE RDH191 TO READ THE S-191 TAPES SENT FROM
C       HOUSTON(JSC). IT THEN LISTS THAT DATA FOR PERUSAL BY SCIENTISTS.
C
C
C       NO1 IN CALL SEQUENCE OF ERRSET, TELLS BEGINNING SYSTEM
C         NUMBER ERROR TO WHICH SUBROUTINE APPLIES
C        NO2  IN CALL SEQ. OF ERRSET, TELLS NUMBER OF ERRORS ALLOWED
C         BEFORE TERMINATION OCCURS.
C        NO3 IN CALL SEQ. OF ERRSET, TELLS NUMBER OF ERROR MESSAGES
C         SYSTEM SHOULD  LIST.
C        NO4  IN CALL SEQ. OF ERRSET, TELLS SYSTEM TO GIVE TRACE FOR
C         EACH ERROR MESSAGE.
C        NO5  IN CALL SEQ. OF ERRSET, TELLS HIGHEST IHC(SYSTEM NO.)
C         NUMBER ERROR TO WHICH SUBROUTINE APPLIES.
C
C
C      NCOND IS AN INDICATOR SUPPLIED BY THE SUBROUTINE RDH191 AND ITS
C       OTHER ENTRY(RD191)      =1 MEANS HOUSEKEEPING DATA
C                               =2 MEANS REGULAR DATA
C                               =3 MEANS END OF FILE
C                               =4 MEANS ERROR - VARIABLE MORE THAN TWO
C                                  BYTES LONG
C         NOTE - WE INITIALIZE IT TO ZERO AS A DEBUG AID
C
C
C       NTYPE IS AN INDICATOR SUPPLIED BY USER =0 MEANS WAVELENGTH IN
C                                                 MICRONS IS DESIRED
C                                                 FOR INDEPENDENT VARIABL
C                                              =1 MEANS WAVENUMBER IN
C                                                 (CM)-1 IS DESIRED
C                                                 FOR INDEPENDENT VARIABL
C       THIS VALUE IS PASSED TO THE SUBROUTINE RDH191 OR
C         ITS OTHER ENTRY(RD191)
C
C
C
      INTEGER NSAMU(21)
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
      INTEGER INDATH(7)/0,0,0,0,0,0,0/
      INTEGER*2 IDATH2( 720),IDATU2( 720),IHEAD(10)
      LOGICAL*1 DATHL(1440),DATUL(1440)
      LOGICAL*1 TITL(8,21),TITU(8)
      INTEGER*2 ITIME(4),ITIME1(3)
      INTEGER IUSED(21) /0,0,0,0,0,0,0,0,0,0,0,0,0,0,0,1,1,1,1,1,1/
      REAL*8 MTYPE(3)
      REAL*8 TYPE(3,2)/ 'WAVELENG','TH IN MI','CRONS   ', 'WAVENUMB',
     1'ER IN (C','M)-1    '/
      DIMENSION DATH( 360),DATU( 360),IDATH( 360),IDATU( 360),D1(6)
      EQUIVALENCE(ITIME(1),ITIME1(1))
      EQUIVALENCE(DATH(1),IDATH(1),IDATH2(1),DATHL(1)),(DATU(1),
     1IDATU(1),IDATU2(1),DATUL(1))
      EQUIVALENCE (INDATH(1),IMM1),(INDATH(2),IMM2),(INDATH(3),IMM3),
     1(INDATH(4),IMM4),(INDATH(5),IMM5),(INDATH(6),IMM6),
     2(INDATH(7),IMM7)
      DATA NO1,NO2,NO3,NO4,NO5/207,20,20,2,999/
      DATA NDAT,NDATH/0,0/
      DATA NCOND/0/
      DATA NTYPE/0/
      EXTERNAL  MYERR
C
C
C     FORMATS
C
 1000 FORMAT(I5)
 2000 FORMAT(1H1,52X,'HOUSEKEEPING DATA SKYLAB 191'/,1X,61X,'PAGE ',I5/,
     11X,5X,'RECORD LENGTH (BYTES)=',I5,14X,'DATE OF TAPE GENERATION=',
     23I3,14X,'START DATE OF DATA=',3I3/,1X,5X,'NO. MEASUREMENTS DESCRIB
     3ED=',I2,10X,'DATA BLOCK LENGTH (BYTES)=',I5,10X,'NO. OF DATA BLOCK
     4S PER DATA RECORD=',I5//,1X,'MEASUREMENT',4X,'START',3X,
     5'NO. SAMPLES',4X,'NO. BITS',6X,'INTERCEPT',5X,'INTERCEPT',7X,
     6'SLOPE',10X,'SLOPE',8X,'FRAME',6X,'TIME'/,1X,13X,'LOCATION',3X,
     7'PER BLOCK',7X,'PER',23X,'COMPONENTS',19X,'COMPONENTS',5X,'TIME',
     86X,'BETWEEN'/,1X,14X,'(BYTES)',15X,'MEASUREMENT',62X,
     9'OFFSET',5X,'SAMPLES'//)
 2050 FORMAT(1H0,2X,8A1,5X,I4,8X,I3,10X,I3,6X,E14.7,1X,I6,1X,I6,1X,
     1E14.7,1X,I6,1X,I6,2X,I8,3X,I8)
 2100 FORMAT(1H1,52X,'HOUSEKEEPING DATA-SKYLAB 191'//,1X,61X,'PAGE ',
     1I5//,1X,32X,60A1//,1X,47X,'DATA IS',1X,3A8/,1X,24X,
     2'FILTER SEGMENT 1',2X,'FILTER SEGMENT 2',2X,'FILTER SEGMENT 3',
     32X,'FILTER SEGMENT4',2X,'FILTER SEGMENT 5',2X,'FILTER SEGMENT 6'/)
 2150 FORMAT(1H0,'NO. ACTIVE WAVELENGTHS',2X,5(6X, I4,8X),6X,I4/)
 2200 FORMAT(1H0,'WAVELENGTH',2X,I5,7X,5(E16.8,2X),E16.8)
 2250 FORMAT(1H0,'NOTE-A VALUE ABOVE OF -1. IS TO BE IGNORED - IT MERELY
```

```
     1 INDICATES THAT WAVELENGTHS FOR THAT SEGMENT HAVE ALREADY BEEN COM
     2PLETED'/)
 2300 FORMAT(1H1,39X,'SKYLAB 191 DATA',2X,'FRAME TIME=',I10,2X,'DR',
     12X,3I3/,1X,61X,'PAGE',1X,I5/)
 2350 FORMAT(1HO,'NEW MEASUREMENT',2X,8A1/)
 2400 FORMAT(1H1,34X,'SKYLAB 191 DATA CONTINUED',2X,'FRAME TIME=',
     1I10,2X,'DR',2X,3I3/,1X,61X,'PAGE',1X,I5/)
 2450 FORMAT(1H ,6E20.8)
 2500 FORMAT(1HO,'NOW HERE'/)
 2550 FORMAT(1HO,'REGULAR DATA RECORD ENCOUNTERED ON FIRST READ - EXPECT
     1ED HOUSEKEEPING - ERROR'/)
 2600 FORMAT(1HO,'WE WERE TOLD TO EXPECT ONLY ONE OR TWO BYTES PER MEASU
     1REMENT PARAMETER - WE HAVE MORE - ERROR***'/)
 2650 FORMAT(1HO,'END OF FILE ENCOUNTERED IN HOUSEKEEPING RECORD - ERROR
     1--****'/)
 2700 FORMAT(1HO,'END OF FILE ENCOUNTERED'/)
C
C
C
C        CALL TO ERRSET MADE SO THAT DUMP WILL BE GIVEN FOR ERROR
C         TERMINATION.
C
C
      CALL ERRSET(NO1,NO2,NO3,NO4,MYERR,NO5)
C
C
C
C       NTYPE IS READ INTO PROGRAM ON UNIT 5
C
      READ(5, 1000) NTYPE
C
C
C       HOUSEKEEPING RECORD - (HEADER RECORD)  READ HERE
C
      CALL RDH191(DATH,NCOND,NTYPE)
      GO TO (10,  250,  270,  260),NCOND
   10 CONTINUE
C
C       HERE, HOUSEKEEPING RECORD IS WRITTEN
C
      NDATH=NDATH+1
C
C
C      THE FOLLOWING REARRANGEMENT OF DATA IS NECESSITATED BY
C       HOUSTON'S METHOD OF STORING DATES. THEY USE THE MILITARY OR
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
C       EUROPEAN METHOD(DAY,MONTH,YEAR), WHILE GODDARD USES
C       THE USUAL LAYMAN-U.S. METHOD(MONTH,DAY,YEAR). ALSO, A
C       SINGLE ARRAY REFERENCE IN IO TAKES LESS TIME THAN SEPARATE
C       LISTINGS.
C
        IHEAD(1)=IDATH2(1)
        IHEAD(2)=IDATH2(3)
        IHEAD(3)=IDATH2(2)
        IHEAD(4)=IDATH2(4)
        IHEAD(5)=IDATH2(6)
        IHEAD(6)=IDATH2(5)
        IHEAD(7)=IDATH2(7)
        IHEAD(8)=IDATH2(8)
        IHEAD(9)=IDATH2(9)
        IHEAD(10)=IDATH2(10)
C
C
        WRITE(9, 2000) NDATH,IHEAD
C
        NMEAS=IDATH2(8)
        IK=310
        IL1=-3
        I21=-5
        I23=3
        I25=4
        I27=5
C
C       THE INDICES ABOVE AND BELOW HAVE THE FOLLOWING
C        CORRESPONDENCES -
C          IK  WILL POINT TO THE INTERCEPT VALUE ASSOCIATED
C              WITH THAT MEASUREMENT
C          IK1 WILL POINT TO THE SLOPE VALUE ASSOCIATED
C              WITH THAT MEASUREMENT
C          IL1 AND IL2 GIVE THE BEGINNING AND ENDING
C              VALUES RESPECTIVELY FOR THE MEASUREMENT
C              TITLE OR NAME IN EBCDIC
C          I21 THRU I22 POINT RESPECTIVELY TO START
C              LOCATION FOR MEASUREMENT,NO. OF SAMPLES FOR
C              IT, AND NO BITS FOR IT.
C          I23 THRU I24 POINT TO THE COMPONENTS OF
C              THE INTERCEPT FOR THAT MEASUREMENT
C          I25 AND I26 POINT RESPECTIVELY TO THE
C              FRAME TIME OFFSET AND TIME BETWEEN SAMPLES
C              FOR THAT MEASUREMENT
C          I27 AND I28 RESPECTIVELY POINT TO THE
```

```
C              COMPONENTS OF THE SLOPE
C
C
      DO     30 LX=1,NMEAS
      IL1=IL1+32
      IL2=IL1+7
      IK=IK+2
      IK1=IK+1
      I21=I21+16
      I22=I21+2
      I23=I23+16
      I24=I23+1
      I25=I25+8
      I26=I25+1
      I27=I27+16
      I28=I27+1
C
C
      WRITE(9, 2050)(DATHL(LZ),LZ=IL1,IL2),(IDATH2(LZ),LZ=I21,I22),
     1DATH(IK) ,(IDATH2(LZ),LZ=I23,I24),DATH(IK1),(IDATH2(LZ),
     2LZ=I27,I28),(DATH(LZ),LZ=I25,I26)
C
C          SAVE THE TITLES OF DATA IN A MORE EASILY ACCESSIBLE
C           LOCATION FOR LATER LISTING WITH REGULAR DATA
C
C
      IKT=IL1-1
      DO     20 LTZ=1,8
      IKT=IKT+1
      TITL(LTZ,LX)=DATHL(IKT)
   20 CONTINUE
C
C          SAVE THE NUMBER OF SAMPLES OF DATA PER
C           MEASUREMENT IN A MORE EASILY ACCESSIBLE ARRAY
C           FOR LATER USE IN LISTING REGULAR DATA
C
      I31=I21+1
      NSAMU(LX)=IDATH2(I31)
   30 CONTINUE
C
C          PRINT MESSAGE INDICATING WHETHER WAVELENGTH
C           OR WAVE NUMBER WAS REQUESTED AND THE UNITS IN
C           WHICH THAT REQUEST WILL BE PRINTED
C
      LT=NTYPE+1
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
      WTYPE(1)=TYPE(1,LT)
      WTYPE(2)=TYPE(2,LT)
      WTYPE(3)=TYPE(3,LT)
      NDATH=NDATH+1
      WRITE(9, 2100) NDATH,(IDATHL(LZ),LZ=693,752),WTYPE
      WRITE(9, 2150) (IDATH2(LZ),LZ=377,382)
C
C          DETERMINE THE HIGHEST NUMBER OF ACTIVE
C           WAVELENGTHS RECORDED FOR ANY ONE FILTER.
C           VALUE WILL BE USED IN LOOP TO PRINT OUT
C           VALUES.
C
      LOOP=IDATH2(377)
      IST=377
      DO    40 IT=1,5
      IST=IST+1
      IF(LOOP.LT.IDATH2(IST)) LOOP=IDATH2(IST)
   40 CONTINUE
C
C
      IMM1=191
      IMM2=IMM1+IDATH2(377)
      IMM3=IMM2+IDATH2(378)
      IMM4=IMM3+IDATH2(379)
      IMM5=IMM4+IDATH2(380)
      IMM6=IMM5+IDATH2(381)
      IMM7=IMM6+IDATH2(382)
C
C          AS ONLY ACTIVE WAVELENGTHS WERE SAVED, VALUES
C           ARE 'BACK TO BACK' - NO ZERO FILL. HOWEVER,
C           WE LIST THEM BY FILTER IN A HORIZONTAL
C           LINE(FIRST ONE FOR FILTER 1,FIRST ONE FOR FILTER 2,
C           FIRST ONE FOR FILTER 3,FIRST ONE FOR FILTER 4,
C           FIRST ONE FOR FILTER 5 AND FIRST ONE FOR FILTER
C           6, FOR EXAMPLE)SO WE INSERT A KEY VALUE OF -1
C           IF ALL ACTIVE WAVELENGTHS FOR THAT FILTER
C           HAVE BEEN LISTED. IT APPEARS THAT 20 WILL
C           BE THE HIGHEST VALUE APPEARING IN ANY ONE FILTER ON
C           ANY ONE TAPE.
C
C           NOTE THAT IMM1,IMM2,ETC. ARE EQUIVALENCED TO
C           ARRAY INDATH!
C
      DO    70 LT=1,LOOP
      DO    60 LZY=1,6
```

```
      I=INDATH(LZY)+LT
      IF(I.GT.INDATH(LZY+1)) GO TO     50
      DI(LZY)=DATH(I)
      GO TO     60
   50 CONTINUE
      DI(LZY)=-1.
   60 CONTINUE
      WRITE(9, 2200) LT,DI
   70 CONTINUE
      WRITE(9, 2250)
C
C          ALL WAVELENGTHS(OR WAVENUMBERS) HAVE BEEN LISTED!
C
C
C          SAVE NUMBER OF ACTIVE WAVELENGTHS FOR EACH DETECTOR
C          FOR EACH FILTER FOR USE IN PRINTING REGULAR DATA
C          NOTE - 3 DETECTORS EXIST FOR FILTER1, 3 DETECTORS
C          FOR FILTER 2, 3 DETECTORS FOR FILTER 3,2 DETECTORS
C          FOR FILTER 4,2 DETECTORS FOR FILTER 5, AND 2 FOR FILTER
C          6 - 15 IN ALL FOR REGULAR DATA THEN 6 MORE
C          MEASUREMENTS.(ALL SKYBET DATA WITH ONE VALUE
C          EACH), ADD TO MAKE THE 21 MEASUREMENTS. NOTE ALSO
C          THAT IUSED WAS INITIALIZED TO 0 FOR THE FIRST 15 AND TO
C          1 FOR THE LAST 6 SO THAT THE LATTER NEED NOT BE
C          SET HERE.
C
      DO 80 LT=1,3
   80 IUSED(LT)=IDATH2(377)
      DO 90 LT=4,6
   90 IUSED(LT)=IDATH2(378)
      DO 100 LT=7,9
  100 IUSED(LT)=IDATH2(379)
      DO 110 LT=10,11
  110 IUSED(LT)=IDATH2(380)
      DO 120 LT=12,13
  120 IUSED(LT)=IDATH2(381)
      DO 130 LT=14,15
  130 IUSED(LT)=IDATH2(382)
  140 CONTINUE
C
C       REGULAR DATA RECORD READ
C
      CALL RD191(DATU,NCOND)
      GO TO (10,150,  280,  260 ),NCOND
  150 CONTINUE
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
C
C       HERE, DATA RECORD IS LISTED
C
C
C
C       SUBROUTINE FRATIM IS CALLED TO CONVERT FRAME TIME IN (10)-4 SEC.
C         UNITS TO HOURS,MINUTES AND SECONDS
C
      CALL FRATIM(IDATU(1),ITIME)
C
C
C
C       PRINTOUT LOGIC HERE IS SET-UP ASSUMING THAT LARGEST 'NO.
C       OF SAMPLES PER MEASUREMENT' WILL BE 20 - HOUSTON  AND
C       EARLIEST TAPES CONFIRMED THIS.
C
      LD1=2
      NSAMPS=0
      DO   240 LZ=1,NMEAS
      IF(LZ.EQ.1) GO TO   160
C
C
C           EVERY 8 MEASUREMENTS, WE START A NEW PAGE WITH
C           ITS OWN TITLE AND NEW NUMBER,(ALL SKYLAB 191
C           DATA PAGES ARE NUMBERED CONSECUTIVELY BY THIS
C           PROGRAM) A CHECK OF MOD(LZ,8) YIELDS NEEDED
C           INFORMATION
C
      IF(MOD(LZ,8).NE.1) GO TO   170
      NDAT=NDAT+1
      WRITE(9, 2400) IDATU(1),ITIME1,NDAT
      GO TO   170
  160 CONTINUE
      NDAT=NDAT+1
      WRITE(9, 2300) IDATU(1),ITIME1,NDAT
  170 CONTINUE
C
C           TITLE OF MEASUREMENT EXTRACTED FOR PRINTING
C           ABOVE MEASUREMENT
C
      DO   180 ILB=1,8
      TITU(ILB)=TITL(ILB,LZ)
  180 CONTINUE
      WRITE(9, 2350) TITU
C
```

```
C          NOTE JOBS PERFORMED BY EACH INDEX:
C          LD1 IS THE INDEX TO THE DATA AS FOUND IN DATU.
C          WE ADD NUMBER OF SAMPLES FOUND FOR THE PREVIOUS
C          MEASUREMENT TO DETERMINE THE START POSITION FOR THE
C          NEW DATA.(PREVIOUS NO. OF SAMPLES  DESIGNATED
C          BY NSAMPS). LD2. THE END OF DATA WHICH IS TO BE
C          PRINTED FOR THIS MEASUREMENT) EQUALS LD1 PLUS THE
C          NUMBER OF 'USEFUL' VALUES,(THOSE ASSOCIATED WITH
C          ACTIVE WAVELENGTHS) MINUS ONE. NSAMP GIVES THE
C          NUMBER OF 'USEFUL' VALUES.
C
      NSAMP=IUSED(LZ)
      LD1=LD1+NSAMPS
C
C          IF NO 'USEFUL' DATA IS FOUND FOR
C          THIS MEASUREMENT,(ASSOCIATED NO. OF ACTIVE
C          WAVELENGTHS IS ZERO), SKIP PRINT PROCESS HERE
C          AND LIST 'NONE HERE'.
C
      IF (NSAMP.EQ.0) GO TO   220
      LD2=LD1+NSAMP-1
      IO=NSAMP/6
      IF(IO.EQ.0) GO TO 200
      ID1=LD1-6
      DO   190 LN=1,IO
      ID1=ID1+6
      ID2=ID1+5
      WRITE(9, 2450) (DATU(LY),LY=ID1,ID2)
  190 CONTINUE
      ID1=ID1+6
      ID2=LD2
      GO TO   210
  200 CONTINUE
      ID1=LD1
      ID2=LD2
  210 CONTINUE
      WRITE(9, 2450) (DATU(LY),LY=ID1,ID2)
      GO TO 230
  220 CONTINUE
      WRITE(9, 2500)
  230 CONTINUE
      NSAMPS=NSAMU(LZ)
  240 CONTINUE
C
C
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
C          READ NEXT RECORD - LISTING OF THIS ONE COMPLETE!
C
C
C
      GO TO 140
C
  250 CONTINUE
C
C      ENCOUNTERED DATA RECORD ON FIRST READ - EXPECTED HOUSEKEEPING
C       ERROR
C
      WRITE(6, 2550)
      CALL ABEND
C
  260 CONTINUE
C
C       ERROR ENCOUNTERED IN READ - POSSIBILITIES -
C       MISUNDERSTANDING BETWEEN HOUSTON AND GODDARD - SUPPOSED ONLY
C       ONE OR TWO BYTES PER RECORD - FOUND MORE!
C       BLOCKSIZE TOO LARGE
C       LOGIC OF INDEX VALUE UPKEEP ON INPUT ARRAY IN ASSEMBLY READ
C       INCORRECT - WRONG VALUE PRESENT
C
      WRITE(6, 2600)
      CALL ABEND
C
  270 CONTINUE
C
C       EARLY UNEXPECTED END OF FILE ENCOUNTERED - ERROR
C
      WRITE(6, 2650)
      RETURN
C
  280 CONTINUE
C
C       NORMAL END OF FILE
C
      WRITE(6, 2700)
      RETURN
      END
       SUBROUTINE FRATIM(IY,ITIME)
C
C         THIS SUBROUTINE WAS WRITTEN BY F.MILASI OF CSC FOR TASK 092,
C         SKYLAB, UNDER CONTRACT NAS-5-11999 IN DEC. '73.
C
```

```
C
C----IT CONVERTS FRAME TIME IN UNITS OF (10)-4 SEC.
C----INTO HOUR, MIN. AND SEC..--------------------
C
C
C---Y=FRAME TIME
C
C
C
      INTEGER*2 ITIME(4)
      REAL*8 X,Y,T,A,B,C
C
C---TO GET THE NUMBER OF HOURS -
C
      Y=IY
      X=Y/10000.
      A=X/3600.
      IA=A
      ITIME(1)=IA
C
C---TO GET THE NUMBER OF MIN. -
C
      T=X-(IA*3600)
      B=T/60
      IB=B
      ITIME(2)=IB
C
C--TO GET THE NUMBER OF SEC. -
C
      C=T-(IB*60)
      C=C+.5
      IC=C
      ITIME(3)=IC
      ITIME(4)=00
      RETURN
      END
      SUBROUTINE MYERR
C
C        THIS SUBROUTINE, WRITTEN BY M.MACK OF CSC FOR TASK
C        092,SKYLAB,UNDER CONTRACT NAS-5-11999, INDEC.'73,IS
C        UTILIZED BY A CALL TO SYSTEM SUBROUTINE ERRSET. THAT
C        ROUTINE PREVENTS SYSTEM ERROR TO TERMINATE PROGRAM,
C        BUT PASSES CONTROL TO MYERR. THEN MYERR CALLS ABEND
C        TO SECURE A DUMP. MANY SYSTEM ERRORS DO NOT TERMINATE
C        WITH A DUMP, SO ERROR TRACING, IF NEEDED, WOULD BE
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
C       DIFFICULT WITHOUT THIS.
C
        CALL ABEND
        RETURN
        END
// EXEC ASMG
//SOURCE.SYSIN DD *
*
*       THIS SUBROUTINE WILL READ SKYLAB S-191 DATA TAPES AND RETURN
*       THE NEEDED VALUES IN A FORTRAN ACCESSIBLE ARRAY FOR FURTHER
*       PROCESSING. IT WAS WRITTEN IN SEPTEMBER,1973 BY M.MACK OF
*       C.S.C. FOR TASK 092 - CONTRACT NAS-5-11999 - SKYLAB.
*       CALL SEQUENCE FOR HOUSEKEEPING ENTRY,(BY WHICH SUBROUTINE
*       IS CALLED), IS RDH191(DATH,NCOND,NTYPE)
*           DATH IS THE OUTPUT ARRAY OF VALUES FROM TAPE
*           NCOND IS AN INDICATOR SUPPLIED TO USER BY SUBROUTINE
*                =1 IT MEANS HOUSEKEEPING DATA PRESENT
*                =2 IT MEANS REGULAR DATA IS PRESENT
*                =3 IT MEANS AN END OF FILE WAS ENCOUNTERED
*                =4 IT MEANS ERROR PRESENT. EITHER VARIABLE WAS MORE
*                   THAN TWO BYTES LONG, OR INPUT DAT 'ARRAY' POINTER
*                   HAS INCORRECT VALUE OR BLOCKSIZE IS INCORRECT
*           NTYPE IS A VARIABLE SUPPLIED BY THE USER - IT INDICATES THE
*           FORM THAT THE USER WISHES THE INDEPENDENT VARIABLE TO TAKE
*                =0 MEANS THE USER WISHES WAVELENGTHS IN MICRONS
*                =1 IT MEANS THE USER WISHES WAVENUMBERS IN (CM)-1
*
*       CALL SEQUENCE FOR REGULAR DATA ACCESSING IS RD191(DATU,NCOND)
*
*
*        THIS MACRO IS USED THROUGHOUT THE PROGRAM
*
*        IT CHANGES A FIXED POINT VALUE TO A FLOATING POINT VALUE
*
*
         MACRO
         FLOAT   &RF,&RC
         ST      &RC,FPT+4
         LPR     &RC,&RC
         O       &RC,=X'46000000'
         ST      &RC,FPT
         LE      &RF,FPT
         AE      &RF,=X'46000000'
         L       &RC,FPT+4
         LTR     &RC,&RC
```

```
          BP        *+6
          LMER      &RF,&RF
          MEND
*
RDH191    CSECT
          ENTRY     RD191
*
*         THIS IS ELIMINATED FOR DEBUGGING PURPOSES
*
*         PRINT     NODATA,NOGEN      ELIMINATES EXTRANEOUS SYSTEM
*                                     GENERATED EXPANSION OF INSTRUCTIONS
*
          SAVE      (14,12),,*        STANDARD SUBROUTINE SET-UP IN WHICH
*                                     SAVE AREAS FROM CALLING PROGRAM ARE
*                                     PRESERVED.
*
*           IN CASE DIFFERENT REGISTERS ARE DESIRED AS PROGRAM EXPANDS,
*           LITERALS ARE SET UP FOR USE IN PROGRAM. THESE CAN BE
*           CHANGED.
*
*             THE FIRST 5, THE SYSTEM REGISTERS, ARE NAMED ONLY
*             TO PROVIDE CLARITY IN UNDERSTANDING THE NAMES GIVEN
*             TO THE OTHERS. WHEN USED, FOR EMPHASIS, EXPLICIT
*             NUMERAL NAMES WILL BE USED.
*
RZ        EQU       0         SYSTEM USED REGISTER
RA        EQU       1         SYSTEM USED REGISTER
RM        EQU       13        SYSTEM USED REGISTER
RN        EQU       14        SYSTEM USED REGISTER
RO        EQU       15        SYSTEM USED REGISTER
RB        EQU       2         USUALLY DATA LOADS
RC        EQU       3         SECONDARY BASE REGISTER
RD        EQU       4         GENERAL USE AND DATA LOADS
RE        EQU       5         LOOP INDEX REGISTER
RF        EQU       6         USED AS INDEX FOR OUTPUT ARRAY
RG        EQU       7         USED AS INDEX FOR GIVEN TAPE INPUT ARRAY
RH        EQU       8         ALSO CALLED DATBASE - SEE BELOW
RI        EQU       9         USED AS BASE REGISTER FOR SUBROUTINE
RJ        EQU       10        INDEX REGISTER
RK        EQU       11        GENERAL USE - SEE CODING
RL        EQU       12        GENERAL USE - SEE CODING
DATBASE   EQU       8         USED AS BASE FOR DATA SAVED SHOULD BE
*                               DIFFERENT FROM BASE REGISTER OF SUBROUTINE
*
*
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
*
*
*
         BALR     R1,0                  GENERAL SUBROUTINE SET-UP
         USING    HERE1,R1,RC
HERE1    L        RC,BASEAD1            THIS ALLOWS USE OF SECOND BASE
*                                       REGISTER
         B        AGAIN1
BASEAD1  DC       A(HERE1+4096)
AGAIN1   ST       13,SAVE+4
         LA       RJ,SAVE
         ST       RJ,8(13)
         LR       13,RJ
*
*           DEBUG
*
         MVI      BUG1,C'1'
         L        DATBASE,0(1)
         USING    DATU,DATBASE          THESE ALLOW REFERENCE TO ARRAY DATU
*                                       IN CODING YET PERMIT DATA TO BE
*                                       PASSED BACK TO CALLING PROGRAM
*
         ST       1,ADDAT               STORE ADDRESS OF DATA PASSED IN CALL
*
*
*           DEBUG
*
         MVI      BUG2,C'2'
         L        RB,4(1)               LOAD RB WITH ADDRESS IN SECOND CALL
*                                       ARGUMENT
         L        RD,0(RB)              PUT THAT VALUE INTO NCOND
         ST       RD,NCOND
         L        RB,8(1)               LOAD RB WITH ADDRESS OF THIRD
*                                       ARGUMENT OF CALL SEQUENCE
         L        RD,0(RB)              STORE VALUE OF THIRD ARGUMENT
*                                       IN NTYPE
         ST       RD,NTYPE
*
*
*
*
*           HOUSEKEEPING DATA READ FIRST
*
         OPEN     (TPS191,(INPUT)) OPEN DATA SET ON THE TAPE
*
```

```
         GET     TPS191              INSERT ADDRESS OF DATA INTO REG. 1
         ST      1,DAT               STORE ADDRESS OF INPUT DATA FOR
*                                    FUTURE REFERENCE
         MVC     DATH(20),0(1)       FIRST PART OF DATA TRANSFERRED AS IS
*                                    NO PROBLEMS HERE
         LH      RB,DATH+14          WE WISH TO SAVE NUMBER OF
*                                    MEASUREMENTS FOR FUTURE USE
         STH     RB,NMEAS+2
*
*
*          NOTE - WE STORE HOUSEKEEPING DATA IN A DIFFERENT ARRAY
*        CALLED DATH, SO CONTENTS CAN BE ACCESSED BY SUBROUTINE IN
*        LATER REGULAR DATA PROCESSING. BEFORE RETURNING TO CALLING
*        PROGRAM, WE TRANSFER IT TO DATU ARRAY,(WHICH THROUGH
*        DATBASE USE ALLOWS IT TO BE GIVEN TO CALLING PROGRAM).
*
*      IN THE FOLLOWING LOOP, DATA IS TRANSFERRED TO OUTPUT ARRAY,(IT
*          IS MODIFIED WHERE NECESSARY). NOTE - ORIGINAL TAPE DOES
*          NOT HAVE SOME FULL WORD DATA ON FULL WORD BOUNDARIES
*
*          ALSO, IN THIS LOOP, WE TREAT ALL AS HALF WORDS, BUT
*          PLACE THEM IN THE PROPER PLACES FOR FORTRAN ACCESSING.
*
*
         LA      RF,20               DATH INDEX
         LA      RG,20               DAT INDEX
         L       RE,NMEAS            LOOP INDEX
HOUSE1   LA      RJ,3                INDEX FOR HOUSE2 LOOP
HOUSE2   LH      RB,0(1,RG)          INSERT DATA FROM TAPE INTO REGISTER
         STH     RB,DATH(RF)         STORE IT IN DATH
         LA      RG,2(RG)            UPDATE DAT INDEX
         LA      RF,2(RF)            UPDATE DATH INDEX
         BCT     RJ,HOUSE2           COMPLETE HOUSE2 LOOP
         SR      RB,RB               INITIALIZE REGISTER TO ZERO
         STH     RB,DATH(RF)         STORE IT IN ARRAY DATH(SPACING TO
*                                    PLACE WORD ON WHOLE WORD BOUNDARY)
         LA      RF,2(RF)            UPDATE DATH INDEX
         LA      RJ,12               INITIALIZE HOUSE3 LOOP INDEX
HOUSE3   LH      RB,0(1,RG)          LOAD HALFWORD FROM DAT INTO REGISTER
         STH     RB,DATH(RF)         STORE IT IN DATH
         LA      RG,2(RG)            UPDATE DAT INDEX
         LA      RF,2(RF)            UPDATE DATH INDEX
         BCT     RJ,HOUSE3           COMPLETE HOUSE3 LOOP
         BCT     RE,HOUSE1           COMPLETE HOUSE1 LOOP
         MVC     DATH+692(60),650(1)
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
         MVC      DATH+752(12),710(1)
*                                      MOVE ALPHANUMERIC TITLE AS WELL AS
*                                      FIGURES FOR NUMBERS OF ACTIVE
*                                      WAVELENGTHS IN EACH FILTER TO DATH
*
*                                      FURTHER NOTE - PROGRAM WAS
*                                      MODIFIED 12/6/73 TO ACCOMODATE
*                                      NEW FORM OF TAPES SENT BY
*                                      HOUSTON, (MORE CLOSELY
*                                      RESEMBLE ORIGINAL DESCRIPTION
*                                      SENT. INSTRUCTION ABOVE
*                                      HAD BEEN -
*                                      MVC    DATH+752(12),712(1)
*                                      SINCE  BYTES 711 AND 712 IN INPUT
*                                      ARRAY ON ACTUAL TAPES HAD
*                                      ZERO FILL INSERTED CONTRARY
*                                      TO HOUSTON DESCRIPTION
*
*          DEBUG
*
         MVI      BUG3,C'3'
*
*
*
*
         LA       RD,6               INITIALIZE INDEX OF FILTER LOOP
         SR       RK,RK              INITIALIZE INDEX FOR DATH WHICH
*                                     POINTS TO NUMBER OF ACTIVE WAVE-
*                                     LENGTHS FOR THAT FILTER
         LA       RF,764             INITIALIZE INDEX OF DATH ARRAY
         LA       RG,722             INITIALIZE INPUT DAT INDEX
*
*                                     HERE,ALSO, MODIFICATION
*                                     WAS MADE 12/6/73 - USED TO
*                                     LOAD RG WITH 724(WHEN ZERO
*                                     FILL HAD BEEN PRESENT)
*
*
         ST       RG,GVAL            TEMPORARILY STORE
*                                     VALUE OF RG IN GVAL FOR
*                                     SUBSEQUENT ACCESSING
*
FILTER   LH       RE,DATH+752(RK)    INITIALIZE LOOP INDEX FOR EACH
*                                     FILTER RESPECTIVELY
         LTR      RE,RE              WE MUST CHECK THAT
```

```
*                                       VALUE IS NOT ZERO
         BZ        ISZERO               IF SO, GO TO ISZERO
         L         RB,NTYPE             CHECK NTYPE TO DETERMINE DESIRED FORM
*                                       OF INDEPENDENT VARIABLE
         LTR       RB,RB
         BZ        WAVELN
*
WAVENO   LH        RB,0(1,RG)           WAVENUMBER DESIRED - LOAD IN GIVEN
*                                       VALUE
         FLOAT     RB,RB                CHANGE INPUT INTEGER VALUE TO
*                                       FLOATING POINT
         DE        RB,=E'10000000.'     CONVERTS INPUT MILLIMICRONS (10-9 M.)
*                                       TO CM.(10-2 M.)
         LE        RD,=E'1.'            LOAD A FLOATING POINT 1. INTO
*                                       REGISTER
         DER       RD,RB                DIVIDE REGISTERS - EFFECTIVELY
*                                       WE HAVE TAKEN RECIPROCAL OF VALUE IN
*                                       RB
         STE       RD,DATH(RF)          STORE VALUE INTO DATH
         LA        RF,4(RF)             UPDATE DATH INDEX
         LA        RG,2(RG)             UPDATE DAT INDEX
         BCT       RE,WAVENO            COMPLETE WAVENO LOOP
         LA        RK,2(RK)             UPDATE INDEX FOR NO. OF
*                                       ACTIVE WAVELENGTHS IN FILTER
*                                       ACCESSING
*
         L         RB,GVAL              LOAD RB WITH LAST VALUE
*                                       IN GVAL
         A         RB,=F'40'            ADD 40 AS THIS IS THE
*                                       NUMBER OF BYTES PROMISED
*                                       IN DESCRIPTION,(TAPES AS
*                                       OF 12/5/73 FOLLOW IT)
         ST        RB,GVAL              STORE CURRENT VALUE FOR
*                                       LATER USE
         LR        RG,RB                LOAD RG WITH THIS VALUE
         BCT       RD,FILTER            COMPLETE FILTER LOOP
         B         FINWV                LOOP IS OVER, SO GO TO FINWV
*                                       TO FINISH PROCESSING
*
WAVELN   LH        RB,0(1,RG)           LOAD GIVEN DATA INTO REGISTER
         FLOAT     RB,RB                CHANGE INPUT INTEGER VALUE TO
*                                       FLOATING POINT
         DE        RB,=E'1000.'         CONVERT INPUT MILLIMICRONS
*                                       (10-9 M.) TO MICRONS(10-6 M.)
         STE       RB,DATH(RF)          STORE RESULT IN OUTPUT ARRAY DATH
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
         LA      RF,4(RF)        UPDATE DATH INDEX
         LA      RG,2(RG)        UPDATE DAT INDEX
         BCT     RE,WAVELN       COMPLETE WAVELN LOOP
ISZERO   LA      RK,2(RK)        UPDATE ACTIVE WAVELENGTHS IN FILTER
*                                ACCESSING INDEX
*
         L       RB,GVAL         LOAD RB WITH LAST VALUE
*                                IN GVAL
         A       RB,=F'40'       ADD 40 AS THIS IS THE
*                                NUMBER OF BYTES PROMISED
*                                IN DESCRIPTION, (TAPES AS
*                                OF 12/5/73 FOLLOW IT)
         ST      RB,GVAL         STORE CURRENT VALUE FOR
*                                LATER USE
         LR      RG,RB           LOAD RG WITH THIS VALUE
         BCT     RD,FILTER       COMPLETE FILTER LOOP
*
*
*
*
*
FINWV    L       RE,NMEAS        INITIALIZE INDEX FOR SLOPE-
*                                INTERCEPT CALCULATION LOOP
         LA      RG,36           INITIALIZE DATH INDEX FOR ACCESSING
*                                OF INTERCEPT AND SLOPE COMPONENT
*                                VALUES
         LA      RF,1244         INITIALIZE DATH INDEX FOR INTERCEPT
*                                SLOPE VALUE OUTPUT
*
COEF2    SR      RJ,RJ           INITIALIZE RJ INDEX TO ZERO
         LA      RL,2            INITIALIZE INNER COEF1 LOOP INDEX
*
COEF1    LR      RK,RJ           TRANSFER CONTENTS OF RJ TO RK
         AR      RK,RG           ADD CONTENTS OF RG - EFFECTIVELY,
*                                THIS GIVES US THE PROPER INDEX FOR
*                                ACCESSING DATH ARRAY FOR INTERCEPT
*                                AND SLOPE COMPONENT VALUES
*                                RESPECTIVELY
         ST      RL,CIND         STORE VALUE OF RL AS REGISTER WILL
*                                BE USED BELOW
         SR      RB,RB           INITIALIZE REGISTER TO ZERO
         LH      RB,DATH(RK)     INSERT COMPONENT(COEFFICIENT)
         LTR     RB,RB           TEST IF VALUE IS NEGATIVE -
*                                HOUSTON JSC RECORDS NEGATIVES
*                                JUST LIKE POSITIVES EXCEPT WITH THE
```

```
*                                       MOST SIGNIFICANT BIT A ONE - IBM
*                                       EXPECTS 2'S COMPLEMENT FOR NEGATIVES
*                                       WE MUST MODIFY NUMBER THEREFORE.
         BP       OK
         N        RB,=X'00007FFF'
         LNR      RB,RB                 CONVERT TO NEGATIVE,(IBM NEGATIVE)
         STH      RB,DATH(RK)           CHANGE VALUE IN DATH SO CORRECT ONE
*                                       IS PRESENT(THEIR NEGATIVE - HOUSTON'S
*                                       - IS DIFFERENT FROM OURS)
OK       BC       0,0                   NO OPERATION - LIKE CONTINUE IN
*                                       FORTRAN
         FLOAT    RB,RB                 CHANGE INTEGER INPUT TO FLOATING
*                                       POINT
         LE       RD,=E'10.'            LOAD REGISTER RD WITH 10.
*                                       IN PREPARATION FOR EXPONENT LOOP
         SR       RL,RL                 INITIALIZE REGISTER TO ZERO
         LH       RL,DATH+2(RK)         LOAD REGISTER WITH EXPONENT COMPONENT
         LTR      RL,RL
*
         BZ       ZERO                  IF ZERO, GO TO ZERO TO PROCESS
*
         BM       NEG                   IF NEGATIVE, GO TO NEG - (EXPONENT
*                                       IS NEGATIVE - CALCULATE VALUE
*                                       DIFFERENTLY)
*
         S        RL,=F'1'              SUBTRACT ONE FROM GIVEN EXPONENT
         BZ       ONE                   IF EXPONENT WERE ONE, POWER IS 10.
*
EXP      ME       RD,=E'10.'            MULTIPLY CONTENTS OF REGISTER BY 10.
         BCT      RL,EXP                CONTINUE TO DO SO UNTIL COMPLETE
*                                       CALCULATION OF GIVEN POWER OF TEN
         B        ONE                   GO TO ONE FOR FURTHER PROCESSING
NEG      N        RL,=X'00007FFF'       TAKE POSITIVE VALUE OF EXPONENT
         LNR      RL,RL
         STH      RL,DATH+2(RK)         CHANGE VALUE IN DATH SO CORRECT ONE
*                                       PRESENT(THEIR NEGATIVE-HOUSTON'S -
*                                       IS DIFFERENT FROM OURS)
         LPR      RL,RL                 TAKE POSITIVE VALUE FOR LOOP INDEX
         S        RL,=F'1'              SUBTRACT ONE
         BZ       MONE                  IF MINUS ONE,GO TO MONE
EXPN     ME       RD,=E'10.'            MULTIPLY CONTENTS OF REGISTER BY 10.
         BCT      RL,EXPN               CONTINUE TO DO SO UNTIL COMPLETE
*                                       CALCULATION OF POWER OF TEN
MONE     STE      RD,FPT                TEMPORARILY STORE REGISTER CONTENTS
         LE       RD,=E'1.'             LOAD REGISTER WITH ONE
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
         DE       RD,FPT              DIVIDE BY VALUE DETERMINED ABOVE AND
*                                     TEMPORARILY STORED.
         B        ONE                 BRANCH TO ONE TO CONTINUE PROCESSING
*
ZERO     LE       RD,=E'1.'           WISH ZERO-TH POWER OF 10 WHICH IS ONE
*
*
ONE      MER      RB,RD               MULTIPLY COEFFICIENT BY POWER OF TEN
         STE      RB,DATH(RF)         STORE RESULT IN OUTPUT ARRAY DATH
         LA       RF,4(RF)            UPDATE OUTPUT INDEX
         LA       RJ,4                UPDATE DISPLACEMENT INDEX FOR COEF1
*                                     LOOP
         L        RL,CIND             RESTORE VALUE OF LOOP INDEX TO RL
         BCT      RL,COEF1            COMPLETE COEF1 LOOP
         LA       RG,32(RG)           UPDATE INDEX FOR COMPONENT ACCESSING
         BCT      RE,COEF2            COMPLETE COEF2 OUTER LOOP
*
*
*
*
*        DEBUG
*
         MVI      BUG4,C'4'
*
*
*                                     WE TRANSFER CONTENTS OF DATH TO DATU
*                                     FOR RETURNING DATA TO CALLING PROGRAM
*
*
*       DROPPING BASE RC HERE PREVENTS AN ERRONEOUS USE OF THAT
*        REGISTER TO REFERENCE DATU. WHEN TRANSFER OF DATA IS
*      COMPLETED, REGISTER WILL BE PICKED UP AGAIN
*
         DROP     RC
*
         MVC      DATU(255),DATH
         MVC      DATU+255(255),DATH+255
         MVC      DATU+510(255),DATH+510
         MVC      DATU+765(255),DATH+765
         MVC      DATU+1020(255),DATH+1020
         MVC      DATU+1275(165),DATH+1275
*
*
         USING    HERE1+4096,RC
*
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
*
*
         SR       RF,RF              ZERO REGISTER
         LH       RF,DATH+18         LOAD REGISTER WITH NUMBER OF BLOCKS
*                                    OF DATA PER PHYSICAL RECORD
         ST       RF,NBLOCK          SAVE FOR FUTURE USE
*
*
*
*
*        FOR FUTURE USE, WE WISH TO CALCULATE THE LENGTH OF
*      USABLE DATA IN EACH PHYSICAL BLOCK
*
*
         SR       RK,RK              ZERO RK
         L        RE,NMEAS           LOAD RE WITH NUMBER OF MEASUREMENTS
         LA       RD,22              SET RD EQUAL TO BEGINNING OF NO.
*                                    OF SAMPLES PER MEASUREMENT DATA
CALCLN   LH       RL,DATH(RD)        LOAD RL WITH NUMBER OF SAMPLES FOR
*                                    THIS MEASUREMENT
         LH       RG,DATH+2(RD)      LOAD RG WITH NO. BITS PER MEAS.
         SR       RF,RF              ZERO RF
         D        RF,=F'8'           DIVIDE NUMBER BY 8
         LR       RJ,RG              PUT RESULT, NO. BYTES, IN RJ
         SR       RF,RF              ZERO RF
         LR       RG,RL              PUT NO. SAMPLES INTO RG
         MR       RF,RJ              MULTIPLY BY NO. BYTES PER MEAS.
         AR       RK,RG              ADD RESULT TO SUB-TOTAL IN RK
         LA       RD,32(RD)          UPDATE MEASUREMENT INDEX
         BCT      RE,CALCLN          FINISH LOOP
         A        RK,=F'4'           ADD FOUR BYTES FOR FRAME TIME
         SR       RJ,RJ              ZERO RJ
         M        RJ,NBLOCK          MULTIPLY BY NO. BLOCKS PER PHYS. REC.
         ST       RK,DATLN           STORE RESULT IN DATLN
*
*
*
*
         L        1,ADDAT            RESTORE ADDRESS OF CALL DATA TO 1
         L        RJ,=F'1'           LOAD NCOND WITH A 1 FOR HOUSEKEEPING
*                                    DATA
         L        RK,4(1)
         ST       RJ,0(RK)
         L        13,4(13)           STANDARD RETURN PROCEDURE
         RETURN   (14,12),T
```

```
*
*
*
*        ------------------------------------------------------------
*
*
*          SECOND AND MORE FREQUENTLY USED ENTRY TO PROGRAM
*
RD191    SAVE    (14,12),,*          ENTRY SET-UP WITH LINKAGE
         BALR    RJ,0
         USING   HERE2,RJ,RC
HERE2    L       RC,BASEAD2          ALLOWS USE OF SECOND BASE REGISTER
         B       AGAIN2
BASEAD2  DC      A(HERE2+4096)
AGAIN2   ST      13,SAVE+4
         LA      RJ,SAVE
         ST      RJ,8(13)
         LR      13,RJ
         L       DATBASE,0(1)
*
*            DEBUG
*
         MVI     BUG5,C'5'
*
         ST      1,ADDAT             SAVE VALUE OF ADDRESS OF ARGUMENTS
*                                    IN CALL SEQUENCE FOR FUTURE USE
         L       RK,4(1)             INSERT NCOND
         L       RJ,0(RK)
         ST      RJ,NCOND
         L       1,DAT               RESTORE ADDRESS OF INPUT DATA
*                                    FROM TAPE TO REGISTER 1
         L       RK,BACK             LOAD RK WITH INDICATOR OF PREVIOUS
*                                    ENTRY TO RD191
         LTR     RK,RK
         BZ      MORDAT              IF ZERO, FIRST ENTRY
*
*          INFORMATION STORED IN DATH FROM HOUSEKEEPING RECORD IS USED
*          IN ACCESSING REGULAR DATA RECORD
*
*
*
KEEPON   L       RK,=F'1'            INDICATE ENTRY TO THIS PORTION
*                                    OF PROGRAM
         ST      RK,BACK             STORE FOR FUTURE USE
*
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
*          DEBUG
*
           MVI     BUG6,C'6'
           L       RJ,NBLOCK          LOAD REGISTER WITH NUMBER OF BLOCKS
*                                     OF DATA LEFT
          LTR      RJ,RJ
           BP      STILLHV            IF POSITIVE, IT MEANS THAT WE STILL
*                                     HAVE DATA
*                                     IN THAT CASE, GO ON TO STILLHV
*
MORDAT     GET     TPS191             FETCH ADDRESS OF INPUT TAPE DATA
           ST      1,DAT              STORE VALUE IN DAT FOR FUTURE USE
*
           MVI     BUG7,C'7'
*
           L       RK,=F'2'           INDICATE ENTRY TO THIS PORTION OF
*                                     PROGRAM
           ST      RK,BACK            STORE FOR FUTURE USE
           SR      RG,RG              INITIALIZE RG INPUT DAT ARRAY
*                                     ACCESSING INDEX TO ZERO
           ST      RG,POSSAVE         RE-INITIALIZE POSSAVE
           SR      RJ,RJ              ZERO REGISTER
           LH      RJ,DATH+18         RE-LOAD NUMBER OF BLOCKS PER DATA
           ST      RJ,NBLOCK          SET(PHYSICAL RECORD) INTO NBLOCK
           B       STORING            BRANCH TO STORING AREA FOR FURTHER
*                                     PROCESSING
*
*
STILLHV    L       RG,POSSAVE         INITIALIZE RG(INDEX OF INPUT DAT
*                                     ARRAY)
*
*
*          DEBUG
*
           MVI     BUG8,C'8'
*                                     TO LAST VALUE USED(LAST ENTRY)
           L       RD,DATLN           LOAD RD WITH LENGTH OF USABLE DATA
*                                     RECORD
           SR      RD,RG              SUBTRACT VALUE OF RG
           BP      STORING            IF THERE IS STILL DATA, GO ON
           B       MORDAT             NEED MORE DATA IN DAT
*
STORING    L       1,DAT              LOAD 1 WITH ADDRESS OF TAPE DATA
           LR      RJ,1
           AR      RJ,RG
           MVC     DATU(4),0(RJ)      INSERT FRAME TIME
*
```

```
*           DEBUG
*
            MVI     BUG9,C'9'
*
*
*
*
*
*
*           LOOP FOR ACCESSING REGULAR DATA - THERE WILL BE
*           A GIVEN NUMBER OF SAMPLES FOR EACH MEASUREMENT - THIS IS
*           FOUND IN THE HOUSEKEEPING RECORD
*
            L       RJ,NMEAS        LOAD LOOP INDEX WITH THE NUMBER OF
*                                   MEASUREMENTS
            LA      RF,4            INITIALIZE  OUTPUT DATU INDEX WITH 4
            LA      RB,20           INITIALIZE INDEX OF DATH ARRAY TO
*                                   BEGINNING OF AREA OF MEASUREMENT
*                                   INFORMATION
            LA      RL,1244         INITIALIZE INDEX IN DATH ARRAY FOR
*                                   SLOPE AND INTERCEPT ACCESSING
*
            LA      RG,4(RG)        ADD FOUR TO BY-PASS FRAME TIME -
*                                   (RECORDED ABOVE)
*
DATLP       ST      RB,DATLPIN      STORE VALUE IN DATLPIN(DATLP INDEX)
            LH      RB,DATH(RB)     LOAD RB WITH EXPECTED BYTE START
*                                   ADDRESS OF DATA
*                                   WE WISH TO CHECK THAT WE ARE
*                                   PROCEEDING PROPERLY
            S       RB,=F'1'        DECREASE BY ONE AS NUMBER IS ACTUAL
*                                   BYTE, NOT DISPLACEMENT TO BYTE.
            LR      RE,RG
            S       RE,POSSAVE      WE WISH POSITION WITH REFERENCE
*                                   TO BEGINNING OF BLOCK(NOT WITH
*                                   RESPECT TO BEGINNING OF PHYSICAL
*                                   RECORD - THIS IS WHAT HOUSTON
*                                   RECORDED IN THE HOUSEKEEPING RECORD.
*
            CR      RE,RB
            BNE     ERROR
            L       RB,DATLPIN      RESTORE VALUE OF INDEX TO RB
*
*           DEBUG
*
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
         MVI     BUGA,C'A'
         LH      RK,DATH+2(RB)       INITIALIZE LOOP INDEX TO
*                                    NUMBER OF SAMPLES PER BLOCK
*                                    OF THE GIVEN MEASUREMENT
         LTR     RK,RK               WE MUST TEST THAT VALUE
*                                    IS NOT ZERO. IF SO,
         BZ      WASZERO             BRANCH TO WASZERO
         SR      RD,RD               ZERO RJ FOR COMING DIVISION
         LH      RE,DATH+4(RB)       LOAD THE NUMBER OF BITS FOR THIS
*                                    MEASUREMENT INTO RK
         D       RD,=F'8'            DIVIDE BY 8 TO GET NUMBER OF BYTES
         LR      RD,RE               RD WILL CONTAIN THE NUMBER OF BYTES
*                                    FOR THIS MEASUREMENT
*                                    NOTE- HOUSTON JSC STATES THAT BITS
*                                    PER MEASUREMENT WILL BE A MULTIPLE
*                                    OF 8,(AND FURTHER THAT IT WILL
*                                    PROBABLY BE 8 OR 16 ONLY)
         S       RD,=F'2'
         BZ      TWOBYTE             IF THERE ARE 2 BYTES PER MEASUREMENT,
*                                    PROCESS DATA AT TWOBYTE
         LR      RD,RE               RELOAD RD WITH NO. BYTES
         S       RD,=F'1'
         BZ      ONEBYTE             IF THERE IS ONE BYTE PER MEASUREMENT
*                                    PROCESS DATA AT ONEBYTE
         B       MORBYTE             ERROR - MORE THAN TWO BYTES - CHECK
*                                    WITH HOUSTON -
*
ONEBYTE  SR      RB,RB               INITIALIZE REGISTER TO ZERO
*
*           DEBUG
*
         MVI     BUGB,C'B'
         IC      RB,0(1,RG)          ACCESS INTEGER PARAMETER FROM DATA
         FLOAT   RB,RB               CHANGE IT(INTEGER) TO FLOATING POINT
         LE      RD,DATH+4(RK)
         LE      RD,DATH+4(RL)       LOAD FLOATING POINT REGISTER RD
*                                    WITH SLOPE,(INDEXED BY FIXED POINT
*                                    REGISTER RL)
         MER     RB,RD               MULTIPLY PARAMETER BY SLOPE
         AE      RB,DATH(RL)         ADD THE INTERCEPT
         STE     RB,DATU(RF)         STORE IN DATA ARRAY
         LA      RG,1(RG)            UPDATE DAT INDEX
         LA      RF,4(RF)            UPDATE DATU INDEX
         BCT     RK,ONEBYTE          CONTINUE SAMPLE LOOP
         LA      RL,8(RL)            UPDATE INDEX IN HOUSEKEEPING WHERE
```

```
*                                       INTERCEPT AND SLOPE ARE ACCESSED
         L       RB,DATLPIN             RESTORE VALUE OF RB FROM DATLPIN
         LA      RB,32(RB)              UPDATE INDEX USED FOR OVERALL HOUSE-
*                                       KEEPING FOR EACH MEASUREMENT
         BCT     RJ,DATLP               COMPLETE MEASUREMENT LOOP
         B       REST                   BRANCH TO AREA WHERE YOU PREPARE
*                                       TO RETURN
*
TWOBYTE  LH      RB,0(1,RG)             ACCESS THE DATA PARAMETER
         LTR     RB,RB
         BP      OKPL                   IF VALUE SENSED IS POSITIVE,
*                                       THERE IS NO DIFFICULTY
         N       RB,=X'0000FFFF'        PROBLEM OCCURS WHEN VALUE
*                                       ON TAPE HAD LEADING BIT 1
*                                       -HOUSTON DID NOT MEAN IT, HERE,
*                                       AS A NEGATIVE-IBM TREATS IT
*                                       AS NEGATIVE IN LOAD, SO WE MUST
OKPL     BC      0,0                    ELIMINATE IBM'S ONES!
*
*          DEBUG
*
         MVI     BUGC,C'C'
         FLOAT   RB,RB                  CHANGE INTEGER INPUT TO FLOATING
*                                       POINT
         LE      RD,DATH+4(RL)          LOAD RD WITH SLOPE
*                                       LOADED INTO FLOATING REGISTER RD
*                                       USING INDEX IN FIXED REGISTER RL
         MER     RB,RD                  MULTIPLY BY SLOPE
         AE      RB,DATH(RL)            ADD THE INTERCEPT
         STE     RB,DATU(RF)            STORE IN ARRAY DATU
         LA      RG,2(RG)               UPDATE THE DAT INDEX
         LA      RF,4(RF)               UPDATE THE OUTPUT DATU INDEX
         BCT     RK,TWOBYTE             CONTINUE SAMPLE LOOP
WASZERO  LA      RL,8(RL)               UPDATE SLOPE-INTERCEPT INDEX IN
*                                       HOUSEKEEPING ARRAY
         L       RB,DATLPIN             RESTORE VALUE OF RB FROM DATLPIN
         LA      RB,32(RB)              UPDATE INDEX USED FOR OVERALL HOUSE-
*                                       KEEPING FOR EACH MEASUREMENT
         BCT     RJ,DATLP               COMPLETE MEASUREMENT LOOP
*
*
*
*
*
*
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
*        END OF MEASUREMENTS - PREPARE TO RETURN
*
*
REST     L       RF,NBLOCK       FETCH NUMBER OF BLOCKS YET TO BE
*                                PROCESSED
         S       RF,=F'1'        SUBTRACT ONE(ONE WAS JUST USED!)
         ST      RF,NBLOCK       STORE BACK IN NBLOCK FOR FUTURE USE
         LR      RD,RG           LOAD CURRENT VALUE OF RG INTO RD
         ST      RD,TEMPSZ       FOR DEBUGGING, STORE VALUE IN TEMPSZ
         S       RD,POSSAVE      SUBTRACT PREVIOUS VALUE
         LH      RE,DATH+16
         CR      RD,RE           COMPARE WITH PROMISED BLOCK SIZE
         BNE     ERROR2          IF NOT EQUAL, THERE IS AN ERROR SOME-
*                                WHERE - CHECK
         L       RJ,=F'2'        LOAD NCOND WITH 2 TO INDICATE REGULAR
*                                DATA
         L       1,ADDAT         RESTORE ADDRESS OF INPUT CALLING
*                                SEQUENCE ARGUMENTS TO 1
         L       RK,4(1)
         ST      RJ,0(RK)
         ST      RG,POSSAVE      STORE CURRENT VALUE OF RG INTO
*                                POSSAVE - IF WHOLE PHYSICAL RECORD
*                                WAS NOT USED, THEN NEXT TIME YOU WILL
*                                ACCESS DATA STARTING AT THAT POSITION
         B       GOBACK          COMPLETE LINKAGE AT GOBACK
*
ERROR2   L       RJ,=F'4'        INDICATES ERROR CONDITION -
*                                IF THIS IS FIRST ENTRY TO
*                                RD191 AND NCOND INDICATES PREVIOUS
*                                DATA WAS NOT HOUSEKEEPING, THIS IS
*                                CONSIDERED AN ERROR
*                                ALSO, AN ERROR IS THOUGHT TO
*                                EXIST IF CURRENT VALUE OF TAPE
*                                DATA INDEX AND ITS PREVIOUS VALUE
*                                DO NOT DIFFER BY THE PROMISED
*                                BLOCKSIZE
         L       1,ADDAT         RESTORE ADDRESS OF INPUT ARGUMENT
*                                LIST TO 1
         L       RK,4(1)
         ST      RJ,0(RK)
         B       GOBACK1         COMPLETE LINKAGE AT GOBACK1
*
ERROR    L       1,ADDAT         RESTORE ADDRESS OF INPUT ARGUMENT
*                                LIST TO 1
         L       RJ,=F'4'
```

```
         L       RK,4(1)
         ST      RJ,0(RK)        INSERT ERROR INDICATOR INTO
*                                MCOND
*                                ERROR HERE CAUSED BY INEQUALITY
*                                BETWEEN RG AND EXPECTED VALUE OF RG)
         B       GOBACK1
*
MORBYTE  L       RJ,=F'4'        VARIABLE WAS MORE THAN 2 BYTES LONG
*                                ERROR
         L       1,ADDAT         RESTORE ADDRESS OF ARGUMENTS IN CALL
         L       RK,4(1)
         ST      RJ,0(RK)
         B       GOBACK1         COMPLETE LINKAGE AT GOBACK1
*
FILEND   L       RJ,=F'3'        END OF FILE - FINISHED
         L       1,ADDAT         RESTORE ADDRESS OF ARGUMENTS IN CALL
         L       RK,4(1)
         ST      RJ,0(RK)
*
*
GOBACK   BC      0,0             NO OPERATION - LIKE CONTINUE IN
*                                FORTRAN - NORMAL EXIT
*                                NO PROBLEMS WERE ENCOUNTERED
*
*
*
*          REINITIALIZE BUGS
*
          MVI    BUG5,C'0'
          MVI    BUG6,C'0'
          MVI    BUG7,C'0'
          MVI    BUG8,C'0'
          MVI    BUG9,C'0'
         MVI     BUGA,C'0'
         MVI     BUGB,C'0'
         MVI     BUGC,C'0'
*
*
GOBACK1  L       13,4(13)        RETURN PROCEDURE - EXIT THRU
*                                HERE INDICATES PROBLEM - DO
*                                NOT WISH TO RE-INITIALIZE
*                                'BUGS' SO CAN FIND THEIR
*                                VALUES IN DUMP.
         RETURN  (14,12),T       RESTORE REGISTERS AND RETURN
*
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
*
*
*          INITIALIZE VARIABLES AND STORAGE HERE
*
*                         LTORG PLACES ALL LITERALS HERE
          LTORG
*
*         FOR DEBUG USE
*
BUG1      DC        C'0'
BUG2      DC        C'0'
BUG3      DC        C'0'
BUG4      DC        C'0'
BUG5      DC        C'0'
BUG6      DC        C'0'
BUG7      DC        C'0'
BUG8      DC        C'0'
BUG9      DC        C'0'
BUGA      DC        C'0'
BUGB      DC        C'0'
BUGC      DC        C'0'
*
*
*
*                 INPUT TAPE DESCRIPTION-
*
TPS191 DCB DSORG=PS,MACRF=(GL),DDNAME=TPS191,EODAD=FILEND,DEVD=TA
*
*
*                   VARIABLES -
*
ADDAT     DC        F'0'          TEMPORARY LOCATION FOR INPUT
*                                 CONTENTS OF 1 - ADDRESS OF CALL
*                                 SEQUENCE
BACK      DC        F'0'          INDICATOR SHOWING WHETHER RD191 HAS
*                                 BEEN USED(ZERO,NO-ONE,YES)
CIND      DC        F'0'          TEMPORARY STORAGE LOCATION FOR
*                                 INDEX REGISTER VALUE
DAT       DC        F'0'          WILL CONTAIN ADDRESS IN CORE OF TAPE
*                                 DATA
DATLN     DC        F'2464'       LENGTH OF USABLE INPUT DAT 'ARRAY'
*                                 FROM TAPE - ACTUAL PHYSICAL BLOCK
*                                 LENGTH IS 2520 BUT THIS INCLUDES
*                                 FILL ZEROS
DATLPIN   DC        F'0'          TEMPORARY STORAGE POSITION FOR INDEX
```

```
*                                       OF DAT ARRAY POINTING TO BEGINNING
*                                       OF AREA OF MEASUREMENT INFORMATION
FPT       DC      2F'0'                 TEMPORARY STORAGE AREA USED BY MACRO
*                                       FLOAT
GVAL      DC      F'0'                  TEMPORARY STORAGE AREA
*                                       FOR CONTENTS OF REG. RG
LENGTH    DC      F'1440'               LENGTH OF OUTPUT ARRAY IN BYTES
NBLOCK    DC      F'-1'                 INITIALIZATION OF NUMBER OF BLOCKS
*                                       LEFT FOR PROCESSING - INDICATES
*                                       NONE WERE DONE AS YET- FIRST ENTRY
NCOND     DC      F'0'                  TEMPORARY STORAGE FOR NCOND - CALL
*                                       SEQUENCE VARIABLE FURNISHED BY THIS
*                                       SUBROUTINE
NMEAS     DC      F'0'                  WILL CONTAIN NUMBER OF MEASUREMENTS
*                                       FOUND ON TAPE - SUPPLIED ON TAPE
NTYPE     DC      F'0'                  TEMPORARY STORAGE FOR INPUT NTYPE
*                                       VARIABLE WHICH INDICATES INDEPENDENT
*                                       VARIABLE DATA PREFERENCE
POSSAVE   DC      F'0'                  INITIALIZATION OF VALUE OF INDEX TO
*                                       ACCESS DAT
SAVE      DC      18F'0'                SAVE AREA FOR LINKAGE
*
TEMPSZ    DC      F'0'                  TEMPORARY STORAGE FOR VALUE OF
*                                       RG FOR DEBUGGING PURPOSES
*
*                 STORAGE-
*
DATH      DS      360F                  OUTPUT ARRAY OF HOUSEKEEPING DATA -
*                                       NEEDED FOR PROCESSING OF REGULAR DATA
DATU      DS      360F                  OUTPUT ARRAY OF REGULAR DATA
*
*
          END
// EXEC LINKGO.REGION.GO=200K
//*    UNIT 6 IS THE DEBUG AND GENERAL MESSAGE UNIT ON PRINTER
//*      UNIT 9 CONTAINS THE LISTING OF THE TAPE CONTENTS
//*      TPS191 IS THE INPUT DATA TAPE
//GO.FT06F001 DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=7265),
// SPACE=(CYL,(7,3))
//GO.FT09F001 DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=7265),
// SPACE=(CYL,(15,5))
//GO.TPS191 DD UNIT=2400-9,DCB=(RECFM=U,BLKSIZE=2520),LABEL=(,BLP),
// VOL=SER=L0001
//GO.SYSUDUMP DD SYSOUT=A
//GO.DATA5 DD *
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

APPENDIX B - NWT191 RUN PROGRAM LISTINGS, INPUT, AND JCL OUTPUT

```
///[illegible]191 JOB (FD0032604D,T,L00360,002H02),JDB,MSGLEVEL=1
//* THISDATE CREATE NEW S-191 TAPES FROM JSC SUPPLIED TAPES
// EXEC FORTRANH,PARM='OPT=2,EBCDIC,LIST,ID,MAP,XREF',REGION=210K
//SOURCE.SYSIN DD *
C
C
C       THIS PROGRAM WAS WRITTEN BY E.PACK OF C.S.C. FOR TASK 092(SKYLAB)
C       UNDER CONTRACT NAS-5 - 11999 IN DECEMBER,1973. IT UTILIZES
C       SUBROUTINE NWT191 TO TAKE THE TWO PAIRED TAPES SENT
C       BY HOUSTON(JSC) AS WELL AS DATA DERIVED FROM INTERPOLATION,
C       (SUBROUTINE ZEMPTN)OF BORESIGHT CAMERA DATA VALUES
C       EXTRACTED FROM BOOKS AND POINTING ANGLES,(EXTRACTED
C       FROM FILMS), TO CREATE A NEW MORE STANDARDIZED AND
C       IBM ACCESSIBLE TAPE OF S-191 VALUES.
C
C
C
C       ISDTPT AND ISDTZN ARE LOGICAL VARIABLES WHICH
C       INDICATE WHETHER DATA EXISTS FOR POINTING
C       ANGLES AND ZENITH ANGLES RESPECTIVELY.
C       IF TRUE, IT DOES EXIST - IF FALSE,
C       IT DOES NOT. HOUSTON JSC CANNOT
C       SUPPLY DATA FOR ALL CASES, SO
C       DR. CURRAN(GSFC) HAS REQUESTED THAT IN THAT
C       CASE, TO SET THE ANGLE OR ANGLES TO ZERO.
C       IN EITHER CASE,(DATA OR NO DATA), THE DATA
C       DELIMITERS OF ( -9) AND (-9) RESPECTIVELY
C       MUST STILL BE INSERTED ON THE DATA5 UNIT(SEE BELOW)
C
C
C      NO1 IN CALL SEQ. OF ERRSET, TELLS BEGINNING SYSTEM
C        NUMBER TO WHICH SUBROUTINE APPLIES
C      NO2 IN CALL SEQ. OF ERRSET, TELLS NUMBER OF ERRORS ALLOWED
C       BEFORE TERMINATION SHOULD OCCUR
C      NO3 IN CALL SEQ. OF ERRSET, TELLS NUMBER OF ERROR
C       MESSAGES SYSTEM SHOULD LIST
C      NO4 IN CALL SEQ. OF ERRSET, TELLS SYSTEM TO GIVE
C       TRACE FOR EACH ERROR MESSAGE
C      NO5 IN CALL SEQ. OF ERRSET, TELLS HIGHEST IHC(SYSTEM NO.)
C       NUMBER ERROR TO WHICH SUBROUTINE APPLIES.
C
C
C
C      NCOND IS AN INDICATOR SUPPLIED BY THE SUBROUTINE NWT191
C                              =1 MEANS END OF FILE
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
C                                   =2 MEANS ERROR ENCOUNTERED
C          NOTE - WE INITIALIZE IT TO ZERO AS A DEBUG AID
C
C
C
C
      REAL*8 TAPE
      LOGICAL ISDTPT,ISDTZN
      COMMON /TAPEN/ TAPE
      COMMON/VALUES/ ITIMPT(3,500),PTING(4,500),ITIMZN(3,500),
     1ZENVL(500),NOPT,NOZE,ISDTPT,ISDTZN
      DIMENSION ITPT(3,500),ITZN(3,500),ZT(500)
      DATA NO1,NO2,NO3,NO4,NO5/207,20,20,2,999/
      DATA NCOND/0/
      EXTERNAL MYERR
C
C
C     FORMATS
C
C
 1000 FORMAT(A8)
 1100 FORMAT(L5)
 1200 FORMAT(I3,1X,I2,1X,3I2,3X,F2.0,3X,F2.0,2X,F2.0,4X,F2.0)
 1300 FORMAT(3I2,F4.3,18X,F4.1,21X,F4.1,5X,I3,2X,I2)
 2000 FORMAT(1H0,'END OF FILE ENCOUNTERED ON CARD ',
     1'INPUT ON UNIT 5 - EXPECTED DELIMITER OF -9!',
     2' POSSIBLE ERROR - ABEND WITH DUMP'/)
 2100 FORMAT(1H0,'END OF FILE ENCOUNTERED ON TAPE')
 2200 FORMAT(1H0,'ERROR IN NWT191')
C
C     THIS ROUTINE,ERRSET, WILL GIVE DUMP IF ERRORS (IHC)
C ARE ENCOUNTERED
C
      CALL ERRSET(NO1,NO2,NO3,NO4,MYERR,NO5)
C
C
C       TAPE UPON WHICH DATA WILL BE PLACED IS READ
C
      READ(5,1000) TAPE
C
C
C         INPUT POINTING ANGLE INFORMATION IS READ
C
      NOPT=0
      READ(5,1100) ISDTPT
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
   10 CONTINUE
      NOPT=NOPT+1
      READ(5,1200,END=120) (ITIMPT(IK,NOPT),IK=1,2),
     1(ITPT(IK,NOPT),IK=1,3),(PTING(IK,NOPT),IK=1,4)
      IF(ITIMPT(1,NOPT).EQ.-9 ) GO TO 20
      GO TO 10
   20 CONTINUE
      NOPT=NOPT-1
C
C     NOW THE ZENITH ANGLE INFORMATION IS READ
C
      NOZE=0
      READ(5,1100) ISDTZN
   30 CONTINUE
      NOZE=NOZE+1
      READ(5,1300,END= 50) (ITZN(IK,NOZE),IK=1,3), ZT(NOZE),
     1ELEV,ZENVL(NOZE),(ITIMZN(IK,NOZE),IK=1,2)
      IF(  ITZN(1,NOZE).EQ. -9 ) GO TO 40
      GO TO 30
   40 CONTINUE
C
      NOZE=NOZE-1
   50 CONTINUE
C
C     NOW ALL VALUES MUST BE TRANSFORMED INTO FORM
C     NEEDED FOR SUBROUTINE ZENPTN
C
C
C          NOTE - TRANSFORMATIONS ARE,NECESSARILY, NOT PERFORMED
C           IF NO DATA EXISTS FOR THAT VARIABLE,(VALUE OF INPUT
C           LOGICAL VARIABLES ,ISDTPT AND ISDTZN, ARE CHECKED
C           FIRST
C
      IF(.NOT. ISDTPT) GO TO 70
      DO  60 LT=1,NOPT
      IH=ITPT(1,LT)
      IM=ITPT(2,LT)
      IS=ITPT(3,LT)
      ITS=IS+IM*60+IH*3600
      ITIMPT(3,LT)=ITS*10000
   60 CONTINUE
C
   70 CONTINUE
      IF(.NOT. ISDTZN) GO TO  90
      DO  80 LT=1,NOZE
```

```
      IH=ITZP(1,LT)
      IM=ITZP(2,LT)
      IS=ITZP(3,LT)
      ITS=IS+IM*60+IH*3600
      SM=ZT(LT)*10000.
      ISM=SM
      ITMZN(3,LT)=ITS*10000+ISM
   80 CONTINUE
   90 CONTINUE
C
C      SUBROUTINE NWT191 IS NOW CALLED
C      IT WILL CREATE THE NEW TAPE FROM THE TWO INPUT TAPES
      CALL NWT191(NCOND)
      GO TO (100,110),NCOND
  100 CONTINUE
      WRITE(6,2100)
C
C          TAPE IS UNLOADED THRU CALL TO WRITEE
C
      CALL WRITEE
      RETURN
  110 CONTINUE
C
C          ERROR ENCOUNTERED IN READING OF TAPE - INDICATED
C           BY VALUE OF NCOND - IBM DUMP SECURED BY CALL TO
C           ABEND
C
      WRITE(6,2200)
      CALL ABEND
  120 CONTINUE
C
C          ERROR ENCOUNTERED IN READING POINTING ANGLE DATA -
C           NO DATA DELIMITER,(-9),FOUND BEFORE END OF FILE
C           ENCOUNTERED ON UNIT 5 - SECURE IBM DUMP THRU
C           CALL TO ABEND
C
      WRITE(6,2000)
      CALL ABEND
C
      RETURN
      END
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
      SUBROUTINE CONV(YY,MM,DD,JUL)
C
C       THIS SUBROUTINE WAS WRITTEN BY F.VILASI OF CSC FOR
C       TASK 092,SKYLAB, UNDER CONTRACT NAS-5-11999 IN
C       NOVEMBER,1973. IT CONVERTS MONTH AND DAY TO
C       CALENDAY DAY.(TAKING INTO CONSIDERATION THE
C       YEAR - WHETHER IT BE A LEAP YEAR ,THAT IS).
C
      INTEGER TOTAL,YY,DD
      DIMENSION       N(12),M(12),MO(11)
      DATA N/64,68,72,76,80,84,88,92,96,00,04,08/
      DATA M/31,59,90,120,151,181,212,243,273,304,334,365/
C----SETTING MO TO 0
      DO 10 I=1,11
      MO(I)=0
   10 CONTINUE
C-----CHECKS FOR LEAP YEAR
      DO 20 I=1,12
      IF(YY.EQ.N(I)) GO TO 60
   20 CONTINUE
      TOTAL=0
C-----CHECKS FOR JAN.
      IF(MM.EQ.01) GO TO 50
C----CHECKS FOR MONTH
      DO 30 I=1,12
      IF(MM.EQ.  I ) GO TO 40
      TOTAL =M(I)
   30 CONTINUE
   40 JUL=TOTAL+DD
      RETURN
   50 JUL=DD
      RETURN
   60 CONTINUE
      TOTAL=0
C----ADDS ONE DAY FOR LEAP YEAR
      IF(MM.EQ. 1) GO TO 40
      DO 70 I=2,11
      MO(I)=M(I)+1
      IF(MM.EQ.  I ) GO TO 40
      TOTAL=MO(I)
   70 CONTINUE
      GO TO 40
      END
       SUBROUTINE FRATIM(IY,ITIME)
C
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
C       THIS SUBROUTINE WAS WRITTEN BY F.MILASI OF CSC FOR TASK 092,
C       SKYLAB, UNDER CONTRACT NAS-5-11999 IN DEC. '73.
C
C
C-----IT CONVERTS FRAME TIME IN UNITS OF (10)-4 SEC.
C-----INTO HOUR, MIN. AND SEC..----------------------
C
C
C---Y=FRAME TIME
C
C
C
      INTEGER*2 ITIME(4)
      REAL*8 X,Y,T,A,B,C
C
C---TO GET THE NUMBER OF HOURS -
C
      Y=IY
      X=Y/10000.
      A=X/3600.
      IA=A
      ITIME(1)=IA
C
C---TO GET THE NUMBER OF MIN. -
C
      T=X-(IA*3600)
      B=T/60
      IB=B
      ITIME(2)=IB
C
C--TO GET THE NUMBER OF SEC. -
C
      C=T-(IB*60)
      C=C+.5
      IC=C
      ITIME(3)=IC
      ITIME(4)=00
      RETURN
      END
      SUBROUTINE MYERR
C
C       THIS SUBROUTINE, WRITTEN BY M.MACK OF CSC FOR TASK
C       092,SKYLAB,UNDER CONTRACT NAS-5-11999, INDEC.'73,IS
C       UTILIZED BY A CALL TO SYSTEM SUBROUTINE ERRSET. THAT
C       ROUTINE PREVENTS SYSTEM ERROR TO TERMINATE PROGRAM,
```

```
C          BUT PASSES CONTROL TO MYERR. THEN IYERR CALLS ABEND
C          TO SECURE A DUMP. ANY SYSTEM ERRORS DO NOT TERMINATE
C          WITH A DUMP, SO ERROR TRACING, IF NEEDED, WOULD BE
C          DIFFICULT WITHOUT THIS.
C
      CALL ABEND
      RETURN
      END
      SUBROUTINE WRITIT(DATUS,IDATU2)
C
C
C
C          THIS SUBROUTINE WAS WRITTEN BY M.MACK OF CSC FOR
C          TASK 092,SKYLAB,UNDER CONTRACT NAS-5-11999 IN
C          DEC. '73. IT IS UTILIZED IN PRODUCTION OF NEW
C          GSFC VERSIONS OF JSC S-191 DATA TAPES. IT
C          UTILIZES FTIO TO MOUNT OUTPUT TAPES, WRITE
C          'NWT191' S-191 OUTPUT DATA ON THOSE TAPES,
C          AND, ALSO, LISTS THAT DATA ON UNIT 9(PRINTER
C          BY JCL DEFINITION), FOR SCIENTISTS' PERUSAL.
C
C
C          IT HAS TWO ENTRIES:
C
C          WRITIT - WITH CALL SEQUENCE(DATUS,IDATU2) WHERE
C            DATUS IS THE REAL*4 ARRAY OF DATA VALUES AND
C            IDATU2 IS THE INTEGER*2 ARRAY OF THOSE SAME
C                  VALUES NEEDED FOR PRINTER OUTPUT.
C          WRITEE - HAS NO ARGUMENTS - IT MERELY UNLOADS
C                  THE OUTPUT TAPE.
C
C
C          ONE COMMON IS PRESENT, TAPEN, WITH ELEMENT TAPE,
C            (REAL*8 VOLUME AND SERIAL NUMBER OF OUTPUT
C            TAPE IN EBCDIC, NEEDED BY FTIO FOR MOUNTING
C            AND UNLOADING.
C
C
C
      INTEGER*2 IDATU2(896)
      LOGICAL HERE/.TRUE./
      COMMON /TAPEN/TAPE
      DIMENSION DATUS(448),DATWRT(8)
C
C      FORMAT STATEMENTS
```

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

```
C
 2000 FORMAT(1H1,14X,'NEW SPECTRUM', 5X,'TIME=',3I3,1X,3I3,
     15X,'LATITUDE=',F6.2,2X,'LONGITUDE=',F6.2/,1X,
     2'ALTITUDE=',F6.1,2X,'YAW=',F6.1,2X,'PITCH=',
     3F6.1,2X,'ROLL=',F6.1,2X,'SUN ZENITH ANGLE=',
     4F6.1,2X,'POINTING ANGLES=',4F8.2/)
 2100 FORMAT(1H0,9X,4('WAVELENGTH',2X,
     1'RADIANCE VALUE',3X)/,10X,4(1X,'(MICRONS)',
     22X,17X)/)
 2200 FORMAT(1H ,9X,4(F10.5,1X,E15.8,3X))
C
C
C       IF FIRST ENTRY, MOUNT TAPE
C
      IF(.NOT. HERE) GO TO 10
      CALL   OUNT(2,12,TAPE,1)
      HERE=.FALSE.
C
   10 CONTINUE
C
C     USING FTIO, WRITE VALUES ON NEW TAPE
C
      CALL FWRITE(DATUS,12,1792)
C
C     NOW,LIST ON OUTPUT UNIT 9 FOR SCIENTISTS PERUSAL
C
      WRITE(9,2000)(IDATU2(IX),IX=1,6),
     1(DATUS(IX),IX=4,14)
      WRITE(9,2100)
C
C        NOTE - IB IS THE INDEX TO DATUS FOR WAVELENGTH
C                  VALUES,(WITH INCREMENTING BELOW, START
C                  INDEX IS ACTUALLY 15)
C               IZ IS THE INDEX TO DATUS FOR RADIANCE
C                  VALUES,(WITH INCREMENTING BELOW, START
C                  INDEX IS ACTUALLY 232)
C
      IB=11
      IZ=228
      DO  30 LR=1,54
      IB=IB+4
      IZ=IZ+4
      IB1=IB-1
      IZ1=IZ-1
C
```

```
C          THIS LOOP ENABLES 4 PAIRS OF WAVELENGTH AND RADIANCE
C          VALUES TO BE PRINTED ON A LINE.
C
C
      DO  20 LK=1,6,2
      LK1=LK+1
      IR1=IR1+1
      IZ1=IZ1+1
      DATWRT(LK)=DATUS(IR1)
      DATWRT(LK1)=DATUS(IZ1)
   20 CONTINUE
      WRITE(9,2200) DATWRT
   30 CONTINUE
C
C
C          AS THE NUMBER OF VALUES IS NOT A MULTIPLE OF 4, THIS
C          LAST LINE WILL NOT HAVE THE FULL FOUR PAIRS, BUT ONLY
C          THE ONE PAIR LISTED BELOW.
C
C
      DATWRT(1)=DATUS(231)
      DATWRT(2)=DATUS(448)
      WRITE(9,2200) (DATWRT(IX),IX=1,2)
      RETURN
C
C
      ENTRY WRITEE
C
C
C          THIS ENTRY, AS NOTED ABOVE, MERELY UNLOADS THE OUTPUT
C          TAPE.
C
C
      CALL UNLOAD(12)
      RETURN
      END
      SUBROUTINE ZENPTN(CALDAY,YEAR,FRTM,ZENITH,PTAN)
C
C  THIS SUBROUTINE WAS WRITTEN BY F.MILASI OF C.S.C. FOR TASK 092(SKYLAB
C  IN DECEMBER,1973. IT WILL TAKE THE ARGUMENTS SENT BY SUBROUTINE
C  'NWT191' AND PERFORM AN INTERPOLATION BASED ON TIME, USING
C  DATA IN ARRAYS FOUND IN THE NAME COMMON 'VALUES'.
C
C   ISDTPT AND ISDTZN ARE LOGICAL VARIABLES WHICH INDICATE WHETHER
C   DATA EXISTS FOR ZENITH ANGLES AND POINTING ANGLES RESPECTIVELY.
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
C    IF TRUE, IT DOES EXIST, IF FALSE, IT DOES NOT. DR. CURRAN (GSFC)
C    HAS REQUESTED THAT IN THAT CASE, TO SET THE ANGLE OR ANGLES TO ZERO.
C
      LOGICAL ISDTPT,ISDTZN,IIDTZN
      INTEGER CALDAY,YEAR,FRTM,TDIF,TDIF1
      COMMON/VALUES/ITIMPT(3,500),PTING(4,500),ITIMZN(3,500),ZENVL(500),
     1NOPT,NOZE,ISDTPT,ISDTZN
      DIMENSION PTAN(4)
      EQUIVALENCE (ITIMZN(1,1),ITZCF),(ITIMZN(2,1),ITZYF),(ITIMZN(3,1),
     1ITZFF)
      DATA LIMIT/600000/
      DATA IIDTZN/.TRUE./
C
      IF(.NOT.ISDTZN)GO TO 185
      J=1
C IS THE INPUT YEAR EARLIER THAN FIRST YEAR OF DATA  IN COMMON DATA ?
      IF(YEAR.LT.ITZYF)CALL ERROR(1,YEAR,CALDAY,FRTM,&190)
C
C IS THE INPUT YEAR LATER THAN FIRST YEAR OF DATA IN ITIMZN COMMON ?
      IF(YEAR.GT.ITZYF)GO TO 10
C
C THE INPUT AND COMMON YEARS ARE EQUAL. IS INPUT CALDAY EARLIER
C THAN DATA IN COMMON ?
      IF(CALDAY.LT.ITZCF)CALL ERROR(2,YEAR,CALDAY,FRTM,&190)
C
C THE INPUT AND COMMON YEARS ARE EQUAL. IS INPUT CALDAY LATER THAN
C DATA IN ITIMZN ?
      IF(CALDAY.GT.ITZCF)GO TO 10
C
C INPUT AND COMMON YEARS ARE EQUAL AND CALDAY ARE EQUAL. IS INPUT FRTM
C EARLIER THAN DATA IN ITIMZN COMMON ?
      IF(FRTM.LT.ITZFF)GO TO 30
C
C INPUT AND COMMON YEARS ARE EQUAL AND DAYS ARE EQUAL. IS FRTM
C EQUAL TO ITIMZN COMMON DATA ?
      IF(FRTM.EQ.ITZFF)GO TO 40
C  IS FRTM LATER THAN FIRST VALUE ?
      IF(FRTM.GT.ITZFF)GO TO 10
C
C   CHECK VALUE OF COUNTER IN COMMON DATA
   10 IF(NOZE.LT.1.OR.NOZE.GT.500)CALL ERROR(7)
C THIS LOOP CHECKS ONLY THOSE VALUES WITH YEAR OR CALDAY GREATER THAN
C ITIMZN COMMON DATA, LOOKING FOR INPUT YEAR AND CALDAY THAT ARE THE
C   SAME IN ITIMZN DATA. THEN CHECKS   INPUT FRTM AGAINST ITIMZN (TIME).
C
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
      DO 20  I=1,NOZE
      J=I
      IF(YEAR.GT.ITIMZN(2,I))GO TO 20
      IF(CALDAY.GT.ITIMZN(1,I))GO TO 20
      IF(FRTM.LT.ITIMZN(3,I))GO TO 100
      IF(FRTM.EQ.ITIMZN(3,I))GO TO 40
       IF(I.EQ.NOZE)GO TO 50
      IF(YEAR.LT.ITIMZN(2,I+1).OR.CALDAY.LT.ITIMZN(1,I+1))GO TO 60
      IF(FRTM.GT.ITIMZN(3,I).AND.FRTM.LT.ITIMZN(3,I+1))GO TO 80
   20 CONTINUE
C
C IF LOOP WAS COMPLETED, INPUT YEAR OR CALDAY WAS GREATER THAN
C    ITIMZN COMMON DATA.      NOW CHECK INPUT TO SEE WHICH WAS
C    GREATER,(YEAR OR CALDAY).
C
      IY=YEAR-ITIMZN(2,NOZE)
      IF(IY.GE.1)CALL ERROR(5,YEAR,CALDAY,FRTM,&190)
      IN=CALDAY-ITIMZN(1,NOZE)
      IF(IN.GE.1)CALL ERROR(4,YEAR,CALDAY,FRTM,&190)
     CALL ERROR(8,YEAR,CALDAY,FRTM,&190)
C
C INPUT AND COMMON YEARS AND CALDAYS ARE THE SAME,AND INPUT FRTM WAS
C EARLIER THAN FIRST ITIMZN DATA (TIME). NOW CHECK TO SEE IF DIFFERENCE
C IS GREATER THAN ONE MINUTE. (OR, CURRAN HAS REQUESTED THAT TEST IN CAS
C INPUT FRTM IS EARLIER THAN FIRST ITIMZN DATA (TIME). CHECK TO SEE IF
C     DIFFERENCE IF GREATER THAN ONE MINUTE, IF SO SET ZENITH EQUAL
C     TO ZERO.
C
   30 IDIFF=ITZERO-FRTM
      IF(IDIFF.GT.LIMIT)CALL ERROR(10,YEAR,CALDAY,FRTM,&190)
      TDIF=ITIMZN(3,J+1)-ITIMZN(3,J)
      IF(TDIF.EQ.0)CALL ERROR(3,YEAR,CALDAY,FRTM,&190)
      ZENITH=ZENVL(J)+((ZENVL(J+1)-ZENVL(J))/TDIF)*(FRTM-ITIMZN(3,J))
      CALL PTING1(CALDAY,YEAR,FRTM,PTAN)
       RETURN
C
C INPUT YEAR AND CALDAY ARE THE SAME AS ITIMZN DATA. INPUT FRTM ALSO
C EQUALS ITIMZN DATA (TIME). SO, SET ZENITH ANGLE TO SAME VALUE AS
C      ITIMZN DATA.
C
   40  ZENITH=ZENVL(J)
       CALL PTING1(CALDAY,YEAR,FRTM,PTAN)
       RETURN
C
C INPUT YEAR AND CALDAY ARE THE SAME AS ITIMZN DATA. INPUT FRTM WAS
```

```
C        GREATER THAN LAST ITIMZN DATA (TIME).          CHECK TO SEE
C        IF DIFFERENCE GREATER THAN ONE MINUTE.
C
   50   TDIF=ITIMZN(3,NOZE)-ITIMZN(3,NOZE-1)
        TDIF1=FRTM-ITIMZN(3,NOZE)
       IF(TDIF1.GT.LIMIT)CALL ERROR(10,YEAR,CALDAY,FRTM,&190)
        GO TO 120
C
C INPUT YEAR AND CALDAY ARE THE SAME AS FIRST ITIMZN DATA, BUT INPUT
C    FRTM IS GREATER THEN FIRST ITIMZN DATA (TIME). THE SECOND TIME
C    IN THE ITIMZN ARRAY HAS CHANGED. (THAT IS THE YEAR IS GREATER
C    THAN THE INPUT YEAR,OR THE CALDAY IS GREATER THAN THE INPUT CALDAY)
C             CHECK TO SEE IF THE YEAR HAS CHANGED AND WHETHER THE
C  ITIMZN DATA (THE SECOND ONE)IS THE FIRST DAY OF THE NEW YEAR.
C
   60   IN=ITIMZN(1,J)-CALDAY
        IF(IN.GE.1)CALL ERROR(4,YEAR,CALDAY,FRTM,&190)
        IY=ITIMZN(2,J)-YEAR
        IF(IY.GE.1)CALL ERROR(5,YEAR,CALDAY,FRTM,&190)
       IY=ITIMZN(2,J+1)-ITIMZN(2,J)
       IF(IY.EQ.0)GO TO 90
       IF(IY.NE.1)CALL ERROR(5,YEAR,CALDAY,FRTM,&190)
       IDAY=365
        IYEAR=ITIMZN(2,J)+1900
       IF(MOD((IYEAR),4).EQ.0)IDAY=366
       IF(ITIMZN(1,J).NE.IDAY)CALL ERROR(6,YEAR,CALDAY,FRTM,&190)
   70   TDIF1=FRTM+(864000000-ITIMZN(3,J))
        IF(TDIF1.GT.LIMIT)CALL ERROR(10,YEAR,CALDAY,FRTM,&190)
        TDIF=ITIMZN(3,J+1)-ITIMZN(3,J)
        IF(TDIF.EQ.0)CALL ERROR(3,YEAR,CALDAY,FRTM,&190)
       ZENITH=ZENVL(J)+((ZENVL(J+1)-ZENVL(J))/TDIF)*TDIF1
       CALL PTING1(CALDAY,YEAR,FRTM,PTAN)
        RETURN
C
C THE INPUT YEAR AND CALDAY ARE THE SAME AS THE FIRST COMMON DATA VALUE,
C AND GREATER THAN FIRST ITIMZN DATA (TIME) AND LESS THAN SECOND
C ITIMZN DATA (TIME). INTERPOLATER TO FIND THE NEAREST ZENITH ANGLE
C BETWEEN THE TWO ITIMZN DATA VALUES.
C
   80   TDIF=ITIMZN(3,J+1)-ITIMZN(3,J)
        ZENITH=ZENVL(J)+((ZENVL(J+1)-ZENVL(J))/TDIF)*(FRTM-ITIMZN(3,J))
        CALL PTING1(CALDAY,YEAR,FRTM,PTAN)
        RETURN
C
C CHECK TO SEE IF CALDAY CHANGED BY ONE DAY. ALSO CHECK
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
C  IF THE TIME DIFFERENCE IS LESS THAN ONE MINUTE BETWEEN THE TIMES OF
C  THE TWO DAYS.
C
   90 IN=ITIMZN(1,J+1)-ITIMZN(1,J)
      IF(IN.EQ.0)CALL ERROR(8,YEAR,CALDAY,FRTM,&190)
      IF(IN.GT.1)CALL ERROR(4,YEAR,CALDAY,FRTM,&190)
       GO TO  70
C
C  INPUT YEAR AND CALDAY ARE THE SAME AS ITIMZN DATA, BUT INPUT FRTM
C     IS LESS THAN ITIMZN DATA (TIME).          CHECK THE TIME
C     BEFORE IT IN THE ITIMZN DATA TO SEE WHETHER THE YEAR
C     HAS CHANGED.
C
  100 IN=ITIMZN(1,J)-CALDAY
      IF(IN.GE.1)CALL ERROR(4,YEAR,CALDAY,FRTM,&190)
      IY=ITIMZN(2,J)-YEAR
      IF(IY.GE.1)CALL ERROR(5,YEAR,CALDAY,FRTM,&190)
       IY=ITIMZN(2,J)-ITIMZN(2,J-1)
       IF(IY.EQ.0) GO TO 130
       IF(IY.NE.1)CALL ERROR(5,YEAR,CALDAY,FRTM,&190)
        IDAY=365
       IYEAR=ITIMZN(2,J-1)+1900
        IF(MOD((    IYEAR    ),4).EQ.0) IDAY=366
        IF(ITIMZN(1,J-1).NE.IDAY)CALL ERROR(9,YEAR,CALDAY,FRTM,&190)
C
  110 TDIF1=FRTM+(864000000-ITIMZN(3,J))
       IF(TDIF1.GT.LIMIT)CALL ERROR(10,YEAR,CALDAY,FRTM,&190)
       TDIF=ITIMZN(3,J)-ITIMZN(3,J-1)
       IF(TDIF.EQ.0)CALL ERROR(3,YEAR,CALDAY,FRTM,&190)
  120 ZENITH=ZENVL(J-1)+((ZENVL(J)-ZENVL(J-1))/TDIF)*TDIF1
       CALL PTING1(CALDAY,YEAR,FRTM,PTAN)
        RETURN
C
C  INPUT CALDAY AND YEAR ARE THE SAME AS ITIMZN DATA, BUT INPUT
C       FRTM IS LESS THAN ITIMZN DATA (TIME).          CHECK THE TIME
C           BEFORE IT IN THE ITIMZN DATA SEE WHETHER THE CALDAY
C       HAS CHANGED.
C
  130  IN=ITIMZN(1,J)-ITIMZN(1,J-1)
       IF(IN.EQ.0)CALL ERROR(8,YEAR,CALDAY,FRTM,&190)
       IF(IN.GT.1)CALL ERROR(4,YEAR,CALDAY,FRTM,&190)
       GO TO 110
C
C       IF ERROR MESSAGE IS CALLED, A MESSAGE WILL BE PRINTED AND
C       ZENITH ANGLE WILL BE SET TO ZERO. (THIS ACTION IS TAKEN
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
C FOR ANY OF THE TIME CHECK 'FAILURES' ON YEAR, CALDAY AND/OR FRTM
C WHETHER  IT FAILED BY BEING TOO EARLY OR TOO LATE.
C SUBROUTINE ERROR PRINTS MESSAGE AND RETURNS TO STATEMENT 190
C WHERE ZENITH IS SET TO ZERO.
C
  185  IF(IIDTZN)CALL ERROR(11,YEAR,CALDAY,FRTM,&190)
  190 ZENITH=0.

      IIDTZN=.FALSE.
       CALL PTING1(CALDAY,YEAR,FRTM,PTAN)
       RETURN
        END
C
       SUBROUTINE PTING1(CALDAY,YEAR,FRTM,PTAN)
C
C
C   THIS SUBROUTINE WAS WRITTEN BY F.MILASI OF C.S.C. FOR TASK 092(SKYLAB
C   IN DECEMBER,1973. IT WILL TAKE THE ARGUMENTS SENT BY SUBROUTINE
C   'ZENPTN' AND PERFORM AN INTERPOLATION BASED ON TIME, USING
C   DATA IN ARRAYS FOUND IN THE NAME COMMON 'VALUES'.
C
C   ISDTPT AND ISDTZN ARE LOGICAL VARIABLES WHICH INDICATE WHETHER
C   DATA EXISTS FOR ZENITH ANGLES AND POINTING ANGLES RESPECTIVELY.
C   IF TRUE, IT DOES EXIST, IF FALSE, IT DOES NOT . DR. CURREN (GSFC)
C   HAS REQUESTED THAT IN THAT CASE, TO SET THE ANGLE OR ANGLES TO ZERO.
C
       LOGICAL ISDTPT,ISDTZN,IIDTPT
        INTEGER CALDAY,YEAR,FRTM,TDIF,TDIF1
        COMMON/VALUES/ITIMPT(3,500),PTING(4,500),ITIMZN(3,500),ZENVL(500)
      1,NOPT,NOZE,ISDTPT,ISDTZN
       DIMENSION PTAN(4)
       EQUIVALENCE (ITIMPT(1,1),ITPCF),(ITIMPT(2,1),ITPYF),(ITIMPT(3,1),

      1ITPFF)
        DATA LIMIT/600000/
        DATA IIDTPT/.TRUE./
C
       IF(.NOT.ISDTPT)GO TO 185
        J=1
C IS THE INPUT YEAR EARLIER THAN FIRST YEAR OF DATA IN COMMON DATA ?
        IF(YEAR.LT.ITPYF)CALL ERROR(1,YEAR,CALDAY,FRTM,&190)
C
C IS THE INPUT YEAR GREATER THAN FIRST YEAR OF DATA IN COMMON DATA ?
        IF(YEAR.GT.ITPYF)GO TO 10
C
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
C  THE INPUT AND COMMON YEARS ARE EQUAL. IS INPUT CALDAY EARLIER
C  THAN DATA IN COMMON ?
      IF(CALDAY.LT.ITPCE)CALL ERROR(2,YEAR,CALDAY,FRTM,&190)
C
C  THE INPUT AND COMMON YEARS ARE EQUAL. IS INPUT CALDAY LATER THAN
C  DATA IN ITIMPT ?
      IF(CALDAY.GT.ITPCE)GO TO 30
C
C  INPUT AND COMMON YEARS ARE EQUAL AND CALDAY ARE EQUAL. IS INPUT FRTM
C  EARLIER THAN DATA IN ITIMPT COMMON ?
      IF(FRTM.LT.ITPFF)GO TO 30
C
C  INPUT AND COMMON YEARS ARE EQUAL AND DAYS ARE EQUAL. IS FRTM
C   EQUAL TO ITIMPT COMMON DATA ?
      IF(FRTM.EQ.ITPFF)GO TO 50
C
C   IS FRTM LATER THAN FIRST VALUE ?
      IF(FRTM.GT.ITPFF)GO TO 10
C
C    CHECK VALUE OF COUNTER IN COMMON DATA
   10 IF(NOPT.LT.1.OR.NOPT.GT.500)CALL ERROR(7,YEAR)
C  THIS LOOP CHECKS ONLY THOSE VALUES WITH YEARS OR CALDAY GREATER THAN
C  ITIMPT COMMON DATA. LOOKING FOR INPUT YEAR AND CALDAY THAT ARE THE
C   SAME IN ITIMPT DATA. THEN CHECKS INPUT FRTM AGAINST ITIMPT (TIME).
C
      DO  20 I=1,NOPT
      J=I
      IF(YEAR.GT.ITIMPT(2,I))GO TO 20
      IF(CALDAY.GT.ITIMPT(1,I))GO TO 20
      IF(FRTM.LT.ITIMPT(3,I))GO TO 140
      IF(FRTM.EQ.ITIMPT(3,I))GO TO 50
      IF(I.EQ.NOPT)GO TO 70
      IF(YEAR.LT.ITIMPT(2,I+1).OR.CALDAY.LT.ITIMPT(1,I+1))GO TO 80
      IF(FRTM.GT.ITIMPT(3,I).AND.FRTM.LT.ITIMPT(3,I+1))GO TO 110

   20 CONTINUE
C
C   IF LOOP WAS COMPLETED, INPUT YEAR OR CALDAY WAS GREATER THAN
C   ITIMPT COMMON DATA. NOW CHECK INPUT TO SEE WHICH WAS
C   GREATER, (YEAR OR CALDAY).
C
      IY=YEAR-ITIMPT(2,NOPT)
      IF(IY.GE.1)CALL ERROR(5,YEAR,CALDAY,FRTM,&190)
      IN=CALDAY-ITIMPT(1,NOPT)
      IF(IN.GE.1)CALL ERROR(4,YEAR,CALDAY,FRTM,&190)
```

```
      CALL ERROR(8,YEAR,CALDAY,ERTM,&190)
C
C   INPUT AND COMMON YEARS AND CALDAYS ARE THE SAME, AND INPUT ERTM WAS
C   EARLIER THAN FIRST ITIMZN DATA (TIME). NOW CHECK TO SEE IF DIFFERENCE
C   IS GREATER THAN ONE MINUTE. (DR. CURRAN HAS REQUESTED THAT TEST IN CA
C   INPUT ERTM IS EARLIER THAN FIRST ITIMPT DATA (TIME). CHECK TO SEE IF
C   DIFFERENCE IF GREATER THAN ONE MINUTE. IF SO SET POINTING EQUAL
C   TO ZERO.
C
   30 TDIFF=ITIMZN-ERTM
      IF(TDIFF.GT.LIMIT)CALL ERROR(10,YEAR,CALDAY,ERTM,&190)
      TDIF=ITIMPT(3,J+1)-ITIMPT(3,J)
      IF(TDIF.EQ.0)CALL ERROR(3,YEAR,CALDAY,ERTM,&190)
      DO  40 L=1,4
       PTAN(L)=PTING(L,J)+((PTING(L,J+1)-PTING(L,J))/TDIF)*
     1(ERTM-ITIMPT(3,J))
   40 CONTINUE

      RETURN
C
C   INPUT YEAR AND CALDAY ARE THE SAME AS ITIMPT DATA. INPUT ERTM ALSO
C  EQUALS ITIMPT DATA (TIME). SO,SET POINTING ANGLES TO SAME VALUE AS
C   ITIMPT DATA.
C
   50  DO 60  L=1,4

       PTAN(L)=PTING(L,J)
   60 CONTINUE
      RETURN
C
C   INPUT YEAR AND CALDAY ARE THE SAME AS ITIMPT DATA. INPUT ERTM WAS
C   GREATER THAN LAST ITIMPT DATA (TIME). CHECK TO SEE IF DIFFERENCE
C  GREATER THAN ONE MINUTE.
C




   70  TDIF=ITIMPT(3,NOPT)-ITIMPT(3,NOPT-1)
       TDIF1=ERTM-ITIMPT(3,NOPT)
      IF(TDIF.GT.LIMIT)CALL ERROR(10,YEAR,CALDAY,ERTM,&190)
       GO TO 160
C
C INPUT YEAR AND CALDAY ARE THE SAME AS FIRST ITIMPT DATA, BUT INPUT
C   ERTM IS GREATER THEN FIRST ITIMPT DATA (TIME). THE SECOND TIME
```


REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
C  IN THE ITIMPT ARRAY HAS CHANGED. (THAT IS THE YEAR IS GREATER
C  THAN THE INPUT YEAR, OR THE CALDAY IS GREATER THAN THE INPUT CALDAY
C  CHECK TO SEE IF THE YEAR HAS CHANGED AND WHETHER THE
C ITIMPT DATA (THE SECOND DATA) IS THE FIRST DAY OF THE NEW YEAR.
C
   80  IN=ITIMPT(1,J)-CALDAY
       IF(IN.GE.1)CALL ERROR(4,YEAR,CALDAY,FRTM,&190)
       IY=ITIMPT(2,J)-YEAR
       IF(IY.GE.1)CALL ERROR(5,YEAR,CALDAY,FRTM,&190)
      IY=ITIMPT(2,J+1)-ITIMPT(2,J)
       IF(IY.EQ.0)GO TO 130
       IF(IY.NE.1)CALL ERROR(5,YEAR,CALDAY,FRTM,&190)
      IDAY=365
       IYEAR=ITIMPT(2,J)+1900
      IF(MOD(ITIMPT(2,J),4).EQ.0)IDAY=366
      IF(ITIMPT(1,J).NE.IDAY)CALL ERROR(6,YEAR,CALDAY,FRTM,&190)
   90  TDIF1=FRTM+(864000000-ITIMPT(3,J))
       IF(TDIF1.GT.LIMIT)CALL ERROR(10,YEAR,CALDAY,FRTM,&190)
       TDIF=ITIMPT(3,J+1)-ITIMPT(3,J)
       IF(TDIF.EQ.0)CALL ERROR(3,YEAR,CALDAY,FRTM,&190)
      DO 100 L=1,4
       PTAM(L)=PTING(L,J)+((PTING(L,J+1)-PTING(L,J))/TDIF)*TDIF1
  100 CONTINUE

       RETURN
C
C  THE INPUT YEAR AND CALDAY ARE THE SAME AS THE FIRST COMMON DATA VALUE
C  AND GREATER THAN FIRST ITIMPT DATA (TIME) AND LESS THAN SECOND
C  ITIMPT DATA (TIME). INTERPOLATER TO FIND THE NEAREST POINTING ANGLE
C  BETWEEN THE TWO ITIMPT DATA VALUES.
C
  110  TDIF=ITIMPT(3,J+1)-ITIMPT(3,J)
      IF(TDIF.EQ.0)CALL ERROR(3,YEAR,CALDAY,FRTM,&190)
       DO 120 L=1,4
       PTAM(L)=PTING(L,J)+((PTING(L,J+1)-PTING(L,J))/TDIF)*
     1(FRTM-ITIMPT(3,J))
  120 CONTINUE

       RETURN
C
C  CHECK TO SEE IF CALDAY CHANGED BY ONE DAY. ALSO CHECK
C  IF THE TIME DIFFERENCE IS LESS THAN ONE MINUTE BETWEEN THE TIMES ON
C  THE TWO DAYS.
C
  130 IN=ITIMPT(1,J+1)-ITIMPT(1,J)
```

```
      IF(IN.EQ.0)CALL ERROR(8,YEAR,CALDAY,FRTM,&190)
      IF(IN.GT.1)CALL ERROR(4,YEAR,CALDAY,FRTM,&190)
       GO TO 90
C
C  INPUT YEAR AND CALDAY ARE THE SAME AS ITIMZN DATA. BUT INPUT FRTM
C  IS LESS THAN ITIMPT DATA (TIME). CHECK THE TIME BEFORE IT IN THE
C ITIMPT DATA TO SEE WHETHER THE YEAR HAS CHANGED.
C
  140 IN=ITIMPT(1,J)-CALDAY
      IF(IN.GE.1)CALL ERROR(4,YEAR,CALDAY,FRTM,&190)
      IY=ITIMPT(2,J)-YEAR
      IF(IY.GE.1)CALL ERROR(5,YEAR,CALDAY,FRTM,&190)
       IY=ITIMPT(2,J)-ITIMPT(2,J-1)
       IF(IY.EQ.0) GO TO 180
       IF(IY.NE.1)CALL ERROR(5,YEAR,CALDAY,FRTM,&190)

        IDAY=365
       IYEAR=ITIMPT(2,J-1)+1900
        IF(MOD(ITIMPT(2,J-1),4).EQ.0) IDAY=366
        IF(ITIMPT(1,J-1).NE.IDAY)CALL ERROR(9,YEAR,CALDAY,FRTM,&190)
  150 TDIF1=FRTM+(86400000-ITIMPT(3,J))
       IF(TDIF1.GT.LIMIT)CALL ERROR(10,YEAR,CALDAY,FRTM,&190)


       TDIF=ITIMPT(3,J)-ITIMPT(3,J-1)
       IF(TDIF.EQ.0)CALL ERROR(3,YEAR,CALDAY,FRTM,&190)
  160 DO 170 L=1,4
      PTAN(L)=PTING(L,J-1)+((PTING(L,J)-PTING(L,J-1))/TDIF)*TDIF1
  170 CONTINUE
        RETURN
C
C  INPUT CALDAY AND YEAR ARE THE SAME AS ITIMPT DATA, BUT INPUT
C  FRTM IS LESS THAN ITIMPT DATA (TIME). CHECK THE TIME
C  BEFORE IT IN THE ITIMPT DATA SEE WHETHER THE CALDAY
C  HAS CHANGED.
C
  180  IN=ITIMPT(1,J)-ITIMPT(1,J-1)
       IF(IN.EQ.0)CALL ERROR(8,YEAR,CALDAY,FRTM,&190)
       IF(IN.GT.1)CALL ERROR(4,YEAR,CALDAY,FRTM,&190)
       GO TO 150
C
C  IF ERROR MESSAGE IS CALLED, A MESSAGE WILL BE PRINTED AND
C POINTING ANGLE WILL BE SET TO ZERO. (THIS ACTION IS TAKEN
C FOR ANY OF THE TIME CHECK 'FAILURES' ON YEAR, CALDAY AND/OR FRTM
C WHETHER IT FAILED BY BEING TOO EARLY OR TOO LATE.
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
C SUBROUTINE ERROR WRITES MESSAGE AND RETURNS TO STATEMENT 190
C WHERE POINTING ANGLES IS SET TO ZERO.
C
  185 IF(IIDTPT)CALL ERROR(12,YEAR,CALDAY,FRTM,&190)
  190 DO 200 L=1,4
      IIDTPT=.FALSE.

      PTAN(L)=0.
  200 CONTINUE
      RETURN
      END
      SUBROUTINE ERROR(IERR,YEAR,CALDAY,FRTM,*)
C
C  THIS SUBROUTINE WAS WRITTEN BY F. MILASI OF C.S.C. FOR TASK 092
C  (SKYLAB) IN DECEMBER, 1973. IT WILL TAKE THE ARGUMENT NUMBER SENT EIT
C  HER BY SUBROUTINE 'ZENPTN' OR SUBROUTINE 'PTING1',WILL PRINT A MESSAG
C  E   RETURN TO STATEMENT 190.THERE POINTING ANGLES OR ZENITH ANGLE
C WILL BE SET TO ZERO. (THIS ACTION IS TAKEN FOR ANY OF THE TIME CHECK
C  'FAILURES' ON YEAR, CALDAY AND/OR FRTM WHETHER    FAILED BY BEING
C  TOO EARLY OR TOO LATE),WHEN COMPARE TO DATA IN COMMON (VALUES).
C
      INTEGER YEAR,CALDAY,FRTM
      DATA MAXERR/12/
      IF(IERR.LT.1.OR.IERR.GT.MAXERR)GO TO 130
      GO TO(10,20,30,40,50,60,70,80,90,100,110,120),IERR
C
C
   10 WRITE(6,2000)IERR,YEAR,CALDAY,FRTM
      RETURN 1
   20 WRITE(6,2050)IERR,YEAR,CALDAY,FRTM
      RETURN 1
   30 WRITE(6,2100)IERR,YEAR,CALDAY,FRTM
      RETURN 1
   40 WRITE(6,2150)IERR,YEAR,CALDAY,FRTM
      RETURN 1
   50 WRITE(6,2200)IERR,YEAR,CALDAY,FRTM
      RETURN 1
   60 WRITE(6,2250)IERR,YEAR,CALDAY,FRTM
      RETURN 1
   70 WRITE(6,2300)IERR,YEAR
      STOP
   80 WRITE(6,2350)IERR,YEAR,CALDAY,FRTM
      RETURN 1
   90 WRITE(6,2400)IERR,YEAR,CALDAY,FRTM
      RETURN 1
```

```
  100 WRITE(6,2450)IERR,YEAR,CALDAY,FRTM
      RETURN 1
  110 WRITE(6,2500)IERR,YEAR,CALDAY,FRTM
      RETURN 1
  120 WRITE(6,2550)IERR,YEAR,CALDAY,FRTM
      RETURN 1
  130 WRITE(6,2600)IERR
      STOP
 2000 FORMAT(1H0,'ERROR NUMBER',I3,2X,'HAS OCCURRED',3X,' INPUT YEAR WAS
     1 LESS THAN FIRST YEAR IN COMMON  (VALUES)',3X,I2,5X,I3.5X,I9)
 2050 FORMAT(1H0,'ERROR NUMBER',I3,2X,'HAS OCCURRED',3X,' INPUT CALENDER
     1 DAY LESS THAN THE FIRST CALENDER DAY IN COMMON (VALUES)',5X,I2,5X
     1,I3,5X,I9)
 2100 FORMAT(1H0,'ERROR NUMBER',I3,2X,'HAS OCCURRED',3X,'TDIF EQUAL ZERO
     1-CHECK COMMON (VALUES) DIVIDING TWO FRAME TIMES THAT ARE THE SAME'
     1,5X,I2,5X,I3.5X,I9)
 2150 FORMAT(1H0,'ERROR NUMBER',I3,2X,'HAS OCCURRED',3X,' INPUT CALENDER
     1 DAY HAS GREATER THAN ANY CALENDER DAY IN COMMON (VALUES)',5X,I2,5
     1X,I3,5X,I9)
 2200 FORMAT(1H0,'ERROR NUMBER',I3,2X,'HAS OCCURRED',3X,' INPUT YEAR WAS
     1 FOUND TO BE GREATER THAN ANY YEAR IN COMMON (VALUES)',5X,I2,5X,I3
     1,5X,I9)
 2250 FORMAT(1H0,'ERROR NUMBER',I3,2X,'HAS OCCURRED',3X,'INPUT CALENDER
     1DAY WAS THE LAST DAY OF YEAR AND SECOND COMMON (VALUES) WAS NOT FI
     1RST DAY OF  NEW YEAR',5X,I2.5X,I3,5X,I9)
 2300 FORMAT(1H0,'ERROR NUMBER',I3,2X,'HAS OCCURRED'3X,' NOZE IS OUT OF
     1 RANGE                    ',I5)
 2350 FORMAT(1H0,'ERROR NUMBER',I3,2X,'HAS OCCURRED',3X,'DATA IN THE COM
     1MON (VALUES) WAS NOT EQUAL TO THE FIRST DAY OF THE NEW YEAR',5X,I2
     1,5X,I3,5X,I9)
 2400 FORMAT(1H0,'ERROR NUMBER',I3,2X,'HAS OCCURRED',3X,'INPUT CALENDER
     1WAS THE FIRST DAY OF NEW YEAR, AND COMMON (VALUES) WAS NOT EQUAL T
     2O (365 OR 366/THAT IS THE CALENDER DAY)',5X,I2,5X,I3,5X,I9)

 2450 FORMAT(1H0,'ERROR NUMBER',I3,' HAS OCCURRED',3X,'TDIF1 WAS GREATER
     1 THAN LIMIT (ONE MINUTE). INPUT FRTM WAS EITHER EARLIER THAN FIRST
     2 COMMON (VALUES) OR LATER THAN LAST COMMON (VALUES).',5X,I2,5X,I3,
     35X,I9)
 2500 FORMAT(1H0,'ERROR NUMBER',I3,1X,'HAS OCCURRED',3X,'NO DATA EXISTS
     1FOR ZENITH ANGLE',9X,I2,5X,I3,5X,I9)
 2550 FORMAT(1H0,'ERROR NUMBER',I3,1X,'HAS OCCURRED',3X,'NO DATA EXISTS
     1FOR POINTING ANGLES',9X,I2,5X,I3,5X,I9)
 2600 FORMAT(1H0,'ERROR NUMBER',I3,1X,'HAS OCCURRED',3X,'IERR GREATER OR
     1 LESS THAN MAXERR')
      END
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
// EXEC ASMG
//SOURCE.SYSIN DD *
*
*       THIS SUBROUTINE WILL READ SKYLAB S-191 DATA TAPES ,(TWO AT
*       A TIME OF TYPES 'A' AND 'B'), AND CREATE A NEW TAPE OF
*       FORTRAN ACCESSIBLE ARRAYS CONTAINING ALL DATA NEEDED FOR
*       ANALYSIS. IT WAS WRITTEN BY M. MACK OF C.S.C. FOR TASK 092
*       - CONTRACT NAS-5-11999 - SKYLAB - IN NOVEMBER AND
*       DECEMBER OF 1973. ITS CALL SEQUENCE IS NWT191(NCOND).
*        (IT NEEDS TO BE CALLED ONLY ONCE)
*            NCOND IS AN INDICATOR SUPPLIED TO USER BY SUBROUTINE
*                  =1 IT MEANS AN END OF FILE WAS ENCOUNTERED
*                  =2 IT MEANS ERROR PRESENT.
*
*
*       HOUSEKEEPING DATA NEEDED IS TEMPORARILY STORED IN
*       ARRAY DATH WHILE REGULAR DATA IS TEMPORARILY
*       STORED IN ARRAY DATU
*
*         THIS MACRO IS USED THROUGHOUT THE PROGRAM
*
*         IT CHANGES A FIXED POINT VALUE TO A FLOATING POINT VALUE
*
*
          MACRO
          FLOAT   &RF,&RC
          ST      &RC,FPT+4
          LPR     &RC,&RC
          O       &RC,=X'46000000'
          ST      &RC,FPT
          LE      &RF,FPT
          AE      &RF,=X'46000000'
          L       &RC,FPT+4
          LTR     &RC,&RC
          BP      *+6
          LNER    &RF,&RF
          MEND
*
NWT191    CSECT
*
*         THIS IS ELIMINATED FOR DEBUGGING PURPOSES
*
*          PRINT   NODATA,NOGEN  *   ELIMINATES EXTRANEOUS SYSTEM
*                                    GENERATED EXPANSION OF INSTRUCTIONS
*
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
         SAVE    (14,12),,*          STANDARD SUBROUTINE SET-UP IN WHICH
*                                    SAVE AREAS FROM CALLING PROGRAM ARE
*                                    PRESERVED.
*
*        IN CASE DIFFERENT REGISTERS ARE DESIRED AS PROGRAM EXPANDS,
*        LITERALS ARE SET UP FOR USE IN PROGRAM. THESE CAN BE
*        CHANGED.
*
*          THE FIRST 5, THE SYSTEM REGISTERS, ARE NAMED ONLY
*          TO PROVIDE CLARITY IN UNDERSTANDING THE NAMES GIVEN
*          TO THE OTHERS. WHEN USED, FOR EMPHASIS, EXPLICIT
*          NUMERAL NAMES WILL BE USED.
*
RZ       EQU     0          SYSTEM USED REGISTERS
RA       EQU     1          SYSTEM USED REGISTERS
RM       EQU     13         SYSTEM USED REGISTERS
RN       EQU     14         SYSTEM USED REGISTERS
RO       EQU     15         SYSTEM USED REGISTERS
RB       EQU     2          USUALLY DATA LOADS
RC       EQU     3          SECONDARY BASE REGISTER
RD       EQU     4          GENERAL USE AND DATA LOADS
RE       EQU     5          LOOP INDEX REGISTER
RF       EQU     6          USED AS INDEX FOR OUTPUT ARRAY
RG       EQU     7          USED AS INDEX FOR GIVEN TAPE INPUT ARRAY
RH       EQU     8          USED AS A THIRD BASE-REGISTER
RI       EQU     9          USED AS BASE REGISTER FOR SUBROUTINE
RJ       EQU     10         INDEX REGISTER
RK       EQU     11         GENERAL USE - SEE CODING
RL       EQU     12         GENERAL USE - SEE CODING
*
*
*
         BALR    RI,0                GENERAL SUBROUTINE SET-UP
         USING   HERE1,RI,RC,RH
HERE1    L       RC,BASEAD1          THIS ALLOWS USE OF SECOND BASE
*                                    REGISTER
         L       RH,BASEAD2
*                                    THIS ALLOWS USE OF A THIRD BASE
*                                    REGISTER
         B       AGAIN1
BASEAD1  DC      A(HERE1+4096)
BASEAD2  DC      A(HERE1+8192)
AGAIN1   ST      13,SAVE+4
         LA      RJ,SAVE
         ST      RJ,8(13)
```

```
         LR      13,RJ
*
*           DEBUG
*
         MVI     BUG1,C'1'
*
         ST      1,ADDAT          STORE ADDRESS OF DATA PASSED IN CALL
*
*
*           DEBUG
*
         MVI     BUG2,C'2'
         L       RB,0(1)          LOAD RB WITH ADDRESS OF CALL
*                                 ARGUMENT
         L       RD,0(RB)         PUT THAT VALUE INTO NCOND
         ST      RD,NCOND
*
*
*
*
*           HOUSEKEEPING DATA READ FIRST
*
         OPEN    (TP191A,(INPUT)) OPEN DATA SET ON THE FIRST TAPE
         GET     TP191A            INSERT ADDRESS OF DATA FROM TAPE
*                                 WHICH IS SUPPOSED TO BE TAPE A INTO
*                                 REGISTER 1
         LH      RJ,710(1)        LOAD 'NUMBER OF ACTIVE WAVELENGTHS IN
*                                 FILTER SEGMENT 1' VALUE INTO RJ
         L       RK,=F'20'        IF THIS IS TAPE A, HOUSTON TAPES
*                                 ORIGINALLY SENT IN JUNE'74 INDICATE
*                                 20 WILL ALWAYS APPEAR ON THIS
*                                 TAPE,(WHICH WE DESIGNATE AS A)
         BNE     TYPEB
TYPEA    ST      1,DATA
         SR      RJ,RJ
         ST      RJ,FINE          FINE IS AN INDICATOR - IF ZERO,
*                                 IT INDICATES THAT A WAS MOUNTED ON A
*                                 AS EXPECTED
         GET     TP191B
         ST      1,DATB
         B       CONTIN
TYPEB    ST      1,DATB
         L       RJ,=F'1'
         ST      RJ,FINE          A VALUE OF 1 IN FINE MEANS TAPES
*                                 WERE MOUNTED ON EACH OTHER'S DRIVES.
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
          GET     TP1910
          ST      1,DATA
*
*
*
CONTIN    L       1,DATA
          MVC     DATH(20),0(1)     FIRST PART OF DATA TRANSFERRED AS IS
*                                   NO PROBLEMS HERE
          LH      RB,DATH+14        WE WISH TO SAVE NUMBER OF
*                                   MEASUREMENTS FOR FUTURE USE
          STH     RB,NMEAS+2
*           NOTE - WE STORE HOUSEKEEPING DATA IN A DIFFERENT ARRAY
*         CALLED DATH, SO CONTENTS CAN BE ACCESSED BY SUBROUTINE IN
*         LATER REGULAR DATA PROCESSING.
*
*     IN THE FOLLOWING LOOP, DATA IS TRANSFERRED TO OUTPUT ARRAY,(IT
*         IS MODIFIED WHERE NECESSARY). NOTE - ORIGINAL TAPE DOES
*         NOT HAVE SOME FULL WORD DATA ON FULL WORD BOUNDARIES
*
*               IN THIS LOOP,WE TREAT ALL AS HALF WORDS, BUT
*         PLACE THEM IN THE PROPER PLACES FOR FORTRAN ACCESSING.
*
*
*
*
          LA      RF,20             DATH INDEX
          LA      RG,20             DAT INDEX
          L       RE,NMEAS          LOOP INDEX
HOUSE1    LA      RJ,3              INDEX FOR HOUSE2 LOOP
HOUSE2    LH      RB,0(1,RG)        INSERT DATA FROM TAPE INTO REGISTER
          STH     RB,DATH(RF)       STORE IT IN DATH
          LA      RG,2(RG)          UPDATE DAT INDEX
          LA      RF,2(RF)          UPDATE DATH INDEX
          BCT     RJ,HOUSE2         COMPLETE HOUSE2 LOOP
          SR      RB,RB             INITIALIZE REGISTER TO ZERO
          STH     RB,DATH(RF)       STORE IT IN ARRAY DATH(SPACING TO
*                                   PLACE WORD ON WHOLE WORD BOUNDARY)
          LA      RF,2(RF)          UPDATE DATH INDEX
          LA      RJ,12             INITIALIZE HOUSE3 LOOP INDEX
HOUSE3    LH      RB,0(1,RG)        LOAD HALFWORD FROM DAT INTO REGISTER
          STH     RB,DATH(RF)       STORE IT IN DATH
          LA      RG,2(RG)          UPDATE DAT INDEX
          LA      RF,2(RF)          UPDATE DATH INDEX
          BCT     RJ,HOUSE3         COMPLETE HOUSE3 LOOP
          BCT     RE,HOUSE1         COMPLETE HOUSE1 LOOP
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
*
*
*
          LH      RJ,DATH+10          WE WILL INSERT TIME INTO OUTPUT ARRAY
          STH     RJ,DATUS
          ST      RJ,DATE             ALSO STORE INTO DATE
          LH      RJ,DATH+8
          STH     RJ,DATUS+2
          ST      RJ,DATE+4
          LH      RJ,DATH+12
          STH     RJ,DATUS+4
          ST      RJ,DATE+8
*
*       SUBROUTINE CONV SUPPLIES JULIAN OR CALENDAR
*       DAY EQUIVALENT OF SUPPLIED MONTH,DAY,YEAR
*
          CALL    CONV,(YEAR,MONTH,DAY,CALDAY)
*                                     FOR USE IN SUBROUTINE
*                                     ZEMPTN, WE CONVERT
*                                     DAY MONTH AND YEAR TO
*                                     CALENDAR DAY
*
*
*
*
          SR      RJ,RJ               INITIALIZE POSAB AND NOINEA INDEX
          LA      RE,12               INITIALIZE LOOP FOR USING ABOVE
*                                     INDEX
*          DEBUG
*
          MVI     BUG3,C'3'
*
          LA      RF,56               INITIALIZE DISPLACEMENT INDEX FOR
*                                     DATUS OUTPUT ARRAY.
WAVE1     L       RG,POSAB(RJ)        LOAD RG WITH DESIRED DISPLACEMENT
*                                     POSITION IN POSAB
          L       RL,NOINEA(RJ)       LOAD RL WITH NUMBER FROM TAPE
*                                     TO BE USED BEFORE ACCESSING
*                                     NEW TAPE.
          LR      RK,RE
          SLL     RK,31               WE WISH TO ISOLATE LOWEST ORDER BIT
          SRL     RK,31
          LTR     RK,RK
          BZ      TPA                 IF ZERO,NUMBER IS EVEN,AND, AS COUNT
*                                     GOES BACKWARDS, EVEN BACKWARD COUNT
```

```
*                                          USE TAPE A.
TPB       L       1,DATB                   LOAD ADDRESS OF B DATA
          B       FURTHER                  CONTINUE PROCESSING
TPA       L       1,DATA                   LOAD ADDRESS OF A DATA
FURTHER   LE      RB,0(1,RG)               LOAD IN THE INPUT WAVELENGTH
*                                          VALUE
          FLOAT   RB,RB                    CONVERT TO FLOATING POINT
          DE      RB,=E'1000.'             CHANGE TO MICRONS(WAS
*                                          MILLIMICRONS)
          STE     RB,DATUS(RF)             STORE IN DATUS
          LA      RF,4(RF)                 UPDATE VALUE OF RF
          LA      RG,2(RG)                 UPDATE VALUE OF RG
          BCT     RL,FURTHER               COMPLETE INNER LOOP
          LA      RJ,4(RJ)                 UPDATE POSAB AND NOINEA
*                                          INDEX
          BCT     RE,WAVE1                 COMPLETE WAVE1 LOOP
*
*
*         NOW WE HAVE ALL WAVELENGTHS RECORDED IN DATUS IN ORDER!
*
*
*
*         NEXT WE CALCULATE SLOPE AND INTERCEPT VALUES
*
*                                          IS ODD FORWARD COUNT, SO EVEN IMPLIES
FINWV     L       RE,NMEAS                 INITIALIZE INDEX FOR SLOPE-
*                                          INTERCEPT CALCULATION LOOP
          LA      RG,36                    INITIALIZE DATH INDEX FOR ACCESSING
*                                          OF INTERCEPT AND SLOPE COMPONENT
*                                          VALUES
          LA      RF,692                   INITIALIZE DATH INDEX FOR INTERCEPT
*                                          SLOPE VALUE OUTPUT
*
COEF2     SR      RJ,RJ                    INITIALIZE RJ INDEX TO ZERO
          LA      RL,2                     INITIALIZE INNER COEF1 LOOP INDEX
*
COEF1     LR      RK,RJ                    TRANSFER CONTENTS OF RJ TO RK
          AR      RK,RG                    ADD CONTENTS OF RG - EFFECTIVELY,
*                                          THIS GIVES US THE PROPER INDEX FOR
*                                          ACCESSING DATH ARRAY FOR INTERCEPT
*                                          AND SLOPE COMPONENT VALUES
*                                          RESPECTIVELY
          ST      RL,CIND                  STORE VALUE OF RL AS REGISTER WILL
*                                          BE USED BELOW
          SR      RB,RB                    INITIALIZE REGISTER TO ZERO
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
         LH      RB,DATH(RK)       INSERT COMPONENT(COEFFICIENT)
         LTR     RB,RB             TEST IF VALUE IS NEGATIVE -
*                                  HOUSTON JSC RECORDS NEGATIVES
*                                  JUST LIKE POSITIVES EXCEPT WITH THE
*                                  MOST SIGNIFICANT BIT A ONE - IBM
*                                  EXPECTS 2'S COMPLEMENT FOR NEGATIVES
*                                  WE MUST MODIFY NUMBER THEREFORE.
         BP      OK
         N       RB,=X'00007FFF'
         LNR     RB,RB             CONVERT TO NEGATIVE.(IBM NEGATIVE)
         STH     RB,DATH(RK)       CHANGE VALUE IN DATH SO CORRECT ONE
*                                  IS PRESENT(THEIR NEGATIVE - HOUSTON'S
*                                  - IS DIFFERENT FROM OURS)
OK       BC      0,0               NO OPERATION - LIKE CONTINUE IN
*                                  FORTRAN
         FLOAT   RB,RB             CHANGE INTEGER INPUT TO FLOATING
*                                  POINT
         LE      RD,=E'10.'        LOAD REGISTER RD WITH 10.
*                                  IN PREPARATION FOR EXPONENT LOOP
         SR      RL,RL             INITIALIZE REGISTER TO ZERO
         LH      RL,DATH+2(RK)     LOAD REGISTER WITH EXPONENT COMPONENT
         LTR     RL,RL
*
         BZ      ZERO              IF ZERO, GO TO ZERO TO PROCESS
*
         BM      NEG               IF NEGATIVE, GO TO NEG - (EXPONENT
*                                  IS NEGATIVE - CALCULATE VALUE
*                                  DIFFERENTLY)
*
         S       RL,=F'1'          SUBTRACT ONE FROM GIVEN EXPONENT
         BZ      ONE               IF EXPONENT WERE ONE, POWER IS 10.
*
EXP      ME      RD,=E'10.'        MULTIPLY CONTENTS OF REGISTER BY 10.
         BCT     RL,EXP            CONTINUE TO DO SO UNTIL COMPLETE
*                                  CALCULATION OF GIVEN POWER OF TEN
         B       ONE               GO TO ONE FOR FURTHER PROCESSING
NEG      N       RL,=X'00007FFF'   TAKE POSITIVE VALUE OF EXPONENT
         LNR     RL,RL
         STH     RL,DATH+2(RK)     CHANGE VALUE IN DATH SO CORRECT ONE
*                                  PRESENT(THEIR NEGATIVE-HOUSTON'S -
*                                  IS DIFFERENT FROM OURS)
         LPR     RL,RL             TAKE POSITIVE VALUE FOR LOOP INDEX
         S       RL,=F'1'          SUBTRACT ONE
         BZ      MONE              IF MINUS ONE,GO TO MONE
EXPN     ME      RD,=E'10.'        MULTIPLY CONTENTS OF REGISTER BY 10.
```

```
         BCT     RL,EXPN          CONTINUE TO DO SO UNTIL COMPLETE
*                                 CALCULATION OF POWER OF TEN
MONE     STE     RD,FPT           TEMPORARILY STORE REGISTER CONTENTS
         LE      RD,=E'1.'        LOAD REGISTER WITH ONE
         DE      RD,FPT           DIVIDE BY VALUE DETERMINED ABOVE AND
*                                 TEMPORARILY STORED.
         B       ONE              BRANCH TO ONE TO CONTINUE PROCESSING
*
ZERO     LE      RD,=E'1.'        WISH ZERO-TH POWER OF 10 WHICH IS ONE
*
*
ONE      MER     RB,RD            MULTIPLY COEFFICIENT BY POWER OF TEN
         STE     RB,DATH(RF)      STORE RESULT IN OUTPUT ARRAY DATH
         LA      RF,4(RF)         UPDATE OUTPUT INDEX
         LA      RJ,4             UPDATE DISPLACEMENT INDEX FOR COEF1
*                                 LOOP
         L       RL,CIND          RESTORE VALUE OF LOOP INDEX TO RL
         BCT     RL,COEF1         COMPLETE COEF1 LOOP
         LA      RG,32(RG)        UPDATE INDEX FOR COMPONENT ACCESSING
         BCT     RE,COEF2         COMPLETE COEF2 OUTER LOOP
*
*        DEBUG
*
         MVI     BUG4,C'4'
*
*
*
*
*
*
         SR      RF,RF            ZERO REGISTER
         LH      RF,DATH+18       LOAD REGISTER WITH NUMBER OF BLOCKS
*                                 OF DATA PER PHYSICAL RECORD
         ST      RF,NBLOCK        SAVE FOR FUTURE USE
*
*
*        FOR FUTURE USE, WE WISH TO CALCULATE THE LENGTH OF
*      USABLE DATA IN EACH PHYSICAL BLOCK
*
*
         SR      RK,RK            ZERO RK
         L       RE,NMEAS         LOAD RE WITH NUMBER OF MEASUREMENTS
         LA      RD,22            SET RD EQUAL TO BEGINNING OF NO.
*                                 OF SAMPLES PER MEASUREMENT DATA
CALCLN   LH      RL,DATH(RD)      LOAD RL WITH NUMBER OF SAMPLES FOR
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
*                                                  THIS MEASUREMENT
            LH      RG,DATH+2(RD)       LOAD RG WITH NO. BITS PER MEAS.
            SR      RE,RE               ZERO RE
            D       RE,=F'8'            DIVIDE NUMBER BY 8
            LR      RJ,RG               PUT RESULT, NO. BYTES, IN RJ
            SR      RE,RE               ZERO RE
            LR      RG,RL               PUT NO. SAMPLES INTO RG
            MR      RE,RJ               MULTIPLY BY NO. BYTES PER MEAS.
            AR      RK,RG               ADD RESULT TO SUB-TOTAL IN RK
            LA      RD,32(RD)           UPDATE MEASUREMENT INDEX
            BCT     RE,CALCLN           FINISH LOOP
            A       RK,=F'4'            ADD FOUR BYTES FOR PRAVE TIME
            SR      RJ,RJ               ZERO RJ
            M       RJ,NBLOCK           MULTIPLY BY NO. BLOCKS PER PHYS. REC.
            ST      RK,DATLN            STORE RESULT IN DATLN
*
*
*
*
*             DEBUG
*
            MVI     BUGS,C'5'
*
            B       MORDAT              FIRST TIME THRU, WE WILL NEED
*                                       TO READ MORE DATA FROM BOTH
*                                       TAPES IMMEDIATELY
*
*
*
*
TRYAGN      L       RJ,NBLOCK           LOAD RJ WITH CURRENT VALUE
*                                       OF NBLOCK
            LTR     RJ,RJ               TEST IF DATA STILL PRESENT
            BP      STILLHV             (NBLOCK WILL STILL BE
*                                       POSITIVE IF THAT IS THE CASE)
*
*
MORDAT      GET     TP191A              FETCH ADDRESS OF INPUT TAPE DATA
*                                       FROM TAPE A
            L       RK,FINE
            LTR     RK,RK               TEST IF ZERO
            BZ      TPOK                IF SO,A IS A AND B IS B!
            ST      1,DATB              DATA TAPES REVERSED - PUT ADDRESS
*                                       OF 'A' IN B AND VICE VERSA
            GET     TP191B
            ST      1,DATA
            B       MORGO
```

```
TPOK     ST      1,DATA          DATA IN PROPER POSITION
         GET     TP191B          SO PUT 'A' IN A AND 'B' IN B
         ST      1,DATB
*
MORGO    BC      0,0             LIKE A 'CONTINUE'- HERE,DATA HAS
*                                JUST BEEN SECURED, SO ALL PERTINENT
*                                INDICATORS MUST BE RE-INITIALIZED
*
*
         MVI     BUG7,C'7'
*
         L       RK,=F'2'        INDICATE ENTRY TO THIS PORTION OF
*                                PROGRAM
         ST      RK,BACK         STORE FOR FUTURE USE
         SR      RG,RG           INITIALIZE RG INPUT DAT ARRAY
*                                ACCESSING INDEX TO ZERO
         ST      RG,POSSAVE      RE-INITIALIZE POSSAVE
         SR      RJ,RJ           ZERO REGISTER
         LH      RJ,DATH+18      RE-LOAD NUMBER OF BLOCKS PER DATA
         ST      RJ,NBLOCK       SET(PHYSICAL RECORD) INTO NBLOCK
         B       STORING         BRANCH TO STORING AREA FOR FURTHER
*                                PROCESSING
*
STILLHV  L       RG,POSSAVE      INITIALIZE RG(INDEX OF INPUT DAT
*                                ARRAY)
*
*          DEBUG
*
         MVI     BUG8,C'8'
*                                TO LAST VALUE USED(LAST ENTRY)
         L       RD,DATLN        LOAD RD WITH LENGTH OF USABLE DATA
*                                RECORD
         SR      RD,RG           SUBTRACT VALUE OF RG
         BP      STORING         IF THERE IS STILL DATA, GO ON
         B       MORDAT          NEED MORE DATA IN DAT
*
*
*
*
STORING  L       1,DATA          LOAD 1 WITH ADDRESS OF TAPE A DATA
         LR      RJ,1
         A       RJ,POSSAVE
         MVC     FRTM(4),0(RJ)   INSERT FRAME TIME
*
*          DEBUG
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
*
            MVI     BUG9,C'9'
*
            CALL    FRATIM,(FRTM,ITIME),VL
            LH      RJ,ITIME            ENTER HOUR,MINUTE, AND
            STH     RJ,DATUS+6          SECOND INTO OUTPUT DATUS
            LH      RJ,ITIME+2          ARRAY - VALUES CALCULATED
            STH     RJ,DATUS+8          IN SUBROUTINE FRATIM
            LH      RJ,ITIME+4
            STH     RJ,DATUS+10
*
*
*
*
*           LOOP FOR ACCESSING REGULAR DATA - THERE WILL BE
*          A GIVEN NUMBER OF SAMPLES FOR EACH MEASUREMENT - THIS IS
*           FOUND IN THE HOUSEKEEPING RECORD
*
*
            LA      RE,12               LOAD LOOP INDEX WITH THE NUMBER OF
*                                       CHANGES OR ENTRIES IN POINTR
            LA      RF,924              INITIALIZE  OUTPUT DATUS INDEX
            SR      RJ,RJ               ZERO RJ - IT WILL BE DISPLACEMENT
*                                       INDEX FOR GRPFILD - WITH 30 ENTRIES
*                                       IT WILL BE UPDATED IN OUTER DATLP1
*                                       LOOP
            SR      RL,RL               ZERO RL - IT WILL BE DISPLACEMENT
*                                       INDEX FOR POINTR - WITH 30 ENTRIES
*                                       IT WILL BE UPDATED IN INNER DATLP2
*                                       LOOP.(ALSO USED FOR 'SLIN')
            ST      RL,RLSAVE           SAVE FOR FUTURE LOAD PURPOSES
DATLP1      SR      RD,RD
            ST      RD,DATLPIND         STORE DISPLACEMENT REFERENCE
*                                       FOR ONE OF 30 'GROUPS'
            L       RD,NOINEA(RJ)       LOAD THE NUMBER OF RADIANCES
*                                       FOR THIS 'GROUP'
*
DATLP2      ST      RD,DATLPIN          STORE CURRENT VALUE OF
*                                       NUMBER OF RADIANCES FOR
*                                       THE 'GROUP' IN DATLPIN
            L       RL,RLSAVE           RE-LOAD THE NEXT RL REFERENCE
            L       RK,GRPFILD(RJ)      LOAD THE NUMBER OF DETECTORS
*                                       RECORDED FOR THIS FILTER AND
*                                       WAVELENGTH
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
         SR        RD,RD
         ST        RD,HAD1
         ST        RD,HAD2
*
DATLP3   L         RG,POINTR(RL)
         A         RG,POSSAVE
         ST        RK,CIND             TEMPORARY STORAGE AS NEED REG. RK
         LR        RB,RE               NOTE - ODD POSITION IN REG. RE WOULD
         SLL       RB,31               BE EVEN IN FORWARD ORDERING,(1-30),
         SRL       RB,31               AND VICE-VERSA - WE CHECK FOR ODD
         LTR       RB,RB               OR EVEN 'QUALITY'
         BZ        NEEDA               NEED DATA FROM 'A' TYPE TAPE
NEEDB    A         RG,DATB             ADD ADDRESS OF 'B' TAPE DATA HERE
         B         DATPRC
NEEDA    A         RG,DATA             ADD ADDRESS OF 'A' TAPE DATA HERE
DATPRC   LR        RB,RG               LOAD RB WITH VALUE IN RG
         A         RB,DATLPIND         ADD CURRENT DISPLACEMENT
         LH        RB,0(RB)            LOAD DATA INTO RB
         LTR       RB,RB
         BP        OKISPOS
*                                      THERE IS NO DIFFICULTY
         N         RB,=X'0000FFFF'     PROBLEM OCCURS WHEN VALUE
*                                      ON TAPE HAD A LEADING BIT
*                                      1-HOUSTON DID NOT MEAN IT, HERE,
*                                      AS A NEGATIVE-IBM TREATS IT
*                                      AS NEGATIVE IN LOAD, SO WE MUST
OKISPOS  BC        0,0                 ELIMINATE IBM'S 1'S.
*
*           DEBUG
*
         MVI       BUGA,C'A'
*
*
REGCASE  BC        0,0
         LTR       RB,RB
         BZ        NOTNEED             IF PARAMETER IS ZERO, WE DON'T
*                                      IT HERE
         FLOAT     RB,RB
         L         RK,SLIN(RL)
         LE        RD,DATH+4(RK)       LOAD THE SLOPE
         MER       RB,RD               MULTIPLY PARAMETER BY IT
         AE        RB,DATH(RK)         ADD THE INTERCEPT
         L         RB,HAD2             LOAD WITH DISPL. FOR TEMP
         STE       RB,TEMP(RB)
         LA        RB,4(RB)
```

```
         ST      RB,HAD2          STORE UPDATED DISPL. INTO HAD2
         L       RK,HAD1
         LA      RK,1(RK)         UPDATE VALUE OF HAD1,
         ST      RK,HAD1          (NO. OF DETEC. USED)
NOTNEED  LA      RL,4(RL)         UPDATE VALUE OF RL,(DISPLACEMENT
*                                 FOR SLIN- SLOPE-INTERCEPT
*                                 ARRAY)
         L       RK,CIND          RESTORE LOOP INDEX VALUE TO
*                                 RK
*
*           DEBUG
*
         MVI     BUGB,C'B'
*
         BCT     RK,DATLP3        COMPLETE DATLP3 LOOP
*
         L       RD,HAD1          CHECK HOW MANY DETEC. HAD
         LTR     RD,RD
         BZ      NOTONE           NONE - ALL ZERO
         C       RD,=F'1'         ONLY ONE CASE ?
         BNE     NRTHONE
JUSTONE  LE      RB,TEMP
*
         STE     RB,DATUS(RF)     STORE IT
         LA      RF,4(RF)         HAVING STORED THE ONE GOOD
*                                 VALUE INTO OUTPUT ARRAY,
*                                 UPDATE INDEX TO IT.
         SER     RB,RB            ZERO RB
         STE     RB,TEMP          RE-INITIALIZE STORAGE AREA
         L       RB,DATLPIND      LOAD DISPLACEMENT FOR
*                                 EACH 'GROUP'
         LA      RB,2(RB)         UPDATE IT
         ST      RB,DATLPIND      STORE IT FOR FUTURE USE
         L       RD,DATLPIN       LOAD RD WITH VALUE OF LOOP
*                                 INDEX PREVIOUSLY SAVED
         BCT     RD,DATLP2        COMPLETE DATLP2 LOOP
         ST      RL,RLSAVE        SAVE VALUE OF RL(LATEST)
         LA      RJ,4(RJ)
         BCT     RE,DATLP1        COMPLETE DATLP1
         B       SKYBET
NOTONE   LE      RB,=E'0.0'       ZERO REG. RB
         STE     RB,DATUS(RF)     STORE ZERO INTO OUTPUT
         LA      RF,4(RF)         UPDATE VALUE OF RF,(DISPLACEMENT
*                                 FOR OUTPUT ARRAY,DATUS)
         SER     RB,RB            ZERO RB AND
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
         STE     RB,TEMP          RE-INITIALIZE STORAGE AREA USING IT
         L       RB,DATLPIND      LOAD DISPLACEMENT FOR
*                                 EACH 'GROUP'
         LA      RB,2(RB)         UPDATE IT
         ST      RB,DATLPIND      STORE IT FOR FUTURE USE
         L       RD,DATLPIN       LOAD RD WITH VALUE OF LOOP
*                                 INDEX PREVIOUSLY SAVED
         BCT     RD,DATLP2        COMPLETE DATLP2 LOOP
         ST      RL,RLSAVE        SAVE VALUE OF RL(LATEST)
         LA      RJ,4(RJ)         UPDATE VALUE OF RJ
         BCT     RE,DATLP1        COMPLETE DATLP1 LOOP
         B       SKYBET
MRTHONE  L       RB,HAD1          LOAD NO. OF CASES FOUND
         LA      RK,4             INITIALIZE RK REG.
         LE      RB,TEMP          LOAD THE FIRST VALUE
         SER     RD,RD            ZERO RD
         STE     RD,TEMP          RE-INITIALIZE STORAGE AREA
*
*           DEBUG
*
         MVI     BUGC,C'C'
*
         S       RB,=F'1'         LOWER NO. CASES BY
*                                 ONE SO AS TO GET
*                                 ADDITION LOOP INDEX
ADDED    AE      RB,TEMP(RK)      ADD NEXT VALUE
         SER     RD,RD            ZERO RD
         STE     RD,TEMP(RK)      RE-INITIALIZE STORAGE AREA
         LA      RK,4(RK)
         BCT     RB,ADDED         COMPLETE ADDITION LOOP
         L       RD,HAD1          RE-LOAD NO. CASES
         FLOAT   RD,RD            CHANGE TO FLOATING POINT
         DER     RB,RD            GET AVERAGE
*
         STE     RB,DATUS(RF)     STORE IN OUTPUT ARRAY
         LA      RF,4(RF)         UPDATE OUTPUT ARRAY
*                                 INDEX
         L       RB,DATLPIND      LOAD DISPLACEMENT FOR
*                                 EACH 'GROUP'
         LA      RB,2(RB)         UPDATE IT
         ST      RB,DATLPIND      STORE IT FOR FUTURE USE
         L       RD,DATLPIN       LOAD RD WITH VALUE OF LOOP
*                                 INDEX PREVIOUSLY SAVED
         BCT     RD,DATLP2        COMPLETE DATLP2 LOOP
         ST      RL,RLSAVE        SAVE VALUE OF RL(LATEST)
```

```
         LA      RJ,4(RJ)          UPDATE GRPFILD INDEX
         BCT     RE,DATLP1         COMPLETE DATLP1
*
*
*
*
*
*
*        WE WILL CALCULATE AND SAVE VALUES OF LATITUDE,LONGITUDE,
*        ALTITUDE, YAW, PITCH AND ROLL HERE
*
SKYRET   L       RG,POSSAVE        LOAD RG WITH PREVIOUS POSITION OF
*                                  DATA POINTER
         A       RG,DATA           ADD ADDRESS OF 'A' TAPE DATA
         A       RG,=F'604'        ADD 604 - DISPL. TO BEGINNING
         LA      RK,812            INITIALIZE RK - REG. FOR
*                                  ACCESSING SLOPE AND INTERCEPT
*                                  VALUES IN DATH
         LH      RB,0(RG)          LOAD LATITUDE PARAMETER
         LTR     RB,RB             TEST IF 'NEGATIVE'
         BP      LTPOS             LEADING BIT ZERO - OK
         BZ      LTPOS
         N       RB,=X'0000FFFF'   SEE EXPLANATION UNDER OKISPOS
LTPOS    BC      0,0               CHANGE TO FLOATING POINT
         FLOAT   RB,RB
         LE      RD,DATH+4(RK)     LOAD SLOPE
         MER     RB,RD             MULTIPLY PARAMETER BY IT
         AE      RB,DATH(RK)       ADD INTERCEPT
         STE     RB,DATUS+12       STORE INTO OUTPUT ARRAY
         LA      RK,8(RK)          UPDATE SLOPE ACCESSING INDEX
         LA      RG,2(RG)          UPDATE DATA ACCESSING INDEX
         LH      RB,0(RG)          LOAD LONGITUDE PARAMETER
         LTR     RB,RB             TEST IF 'NEGATIVE'
         BP      LNPOS             IF LEADING BIT ZERO,OK
         BZ      LNPOS
         N       RB,=X'0000FFFF'   SEE EXPLANATION UNDER OKISPOS
LNPOS    BC      0,0               CHANGE TO FLOATING POINT
         FLOAT   RB,RB
         LE      RD,DATH+4(RK)     LOAD SLOPE
         MER     RB,RD             MULTIPLY PARAMETER BY IT
         AE      RB,DATH(RK)       ADD INTERCEPT
         LTER    RB,RB             TEST FOR SIGN
         BZ      ZLON              HOUSTON GIVES ZERO VALUE
         BM      WLON              HOUSTON HAS W.LONGITUDE AS
*                                  NEGATIVE FOR W. LONG. VALUES
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
*                                                    UP TO 180.
POSLON   LE      RD,=E'360.'       LONGITUDE HERE, FROM
*                                  HOUSTON, IS E. LONGITUDE,
*                                  (VALUES UP TO 180.), BUT GSFC
*                                  WISHES W. LONGITUDE,(180. TO 360.)
         SER     RD,RB             CONVERT IT BY SUBTRACTING FROM
         STE     RD,DATUS+16       360. AND STORE
         B       ALTIT
ZLON     STE     RB,DATUS+16       ZERO VALUE SAME FOR BOTH
*                                  HOUSTON'S NUMBERING AND OURS.
         B       ALTIT
WLON     ME      RB,=E'-1.'        WISH POSITIVE EQUIVALENT
         STE     RB,DATUS+16       STORE IN OUTPUT ARRAY
ALTIT    LA      RG,2(RG)          UPDATE DATA ACCESSING INDEX
         LA      RK,8(RK)          UPDATE SLOPE-INTERCEPT ACCESSING
*                                  INDEX
*
*
*
         LA      RE,4              INITIALIZE LOOP INDEX FOR
*                                  DETERMINING ALTITUDE,YAW,PITCH,
*                                  AND ROLL
         LA      RF,20             INITIALIZE OUTPUT ARRAY INDEX
ALYPR    LH      RB,0(RG)          LOAD IN PARAMETER
         LTR     RB,RB             TEST IF 'NEGATIVE'
         BP      OKVAL
         BZ      OKVAL              OK IF LEADING BIT ZERO
         N       RB,=X'0000FFFF'   SEE EXPLANATION UNDER OKISPOS
OKVAL    BC      0,0               CHANGE TO FLOATING POINT
         FLOAT   RB,RB
         LE      RD,DATH+4(RK)     LOAD SLOPE
         MER     RB,RD             MULTIPLY PARAMETER BY IT
         AE      RB,DATH(RK)       ADD INTERCEPT
         STE     RB,DATUS(RF)      STORE IN OUTPUT ARRAY
         LA      RF,4(RF)          UPDATE OUTPUT ARRAY INDEX
         LA      RK,8(RK)          UPDATE SL.-INTER. INDEX
         LA      RG,2(RG)          UPDATE DATA ACCESS INDEX
         BCT     RE,ALYPR          COMPLETE LOOP
*
*
*
*
*
*
*
```

```
*       WE NEED NOW ONLY CALL SUBROUTINE ZENPTN  TO DERIVE
*       VALUES OF SUN ZENITH ANGLE AND ALSO THE POINTING ANGLES
*
        CALL    ZENPTN,(CALDAY,YEAR,FRTM,ZENITH,PTAN)
*
        LE      RB,ZENITH       LOAD THE ZENITH ANGLE
        STE     RB,DATUS+36     STORE IN OUTPUT ARRAY
        LE      RB,PTAN         LOAD FIRST ANGLE
        STE     RB,DATUS+40     STORE IN OUTPUT ARRAY
        LE      RB,PTAN+4       LOAD SECOND ANGLE
        STE     RB,DATUS+44     STORE IN OUTPUT ARRAY
        LE      RB,PTAN+8       LOAD THIRD ANGLE
        STE     RB,DATUS+48     STORE IN OUTPUT ARRAY
        LE      RB,PTAN+12      LOAD IN FOURTH ANGLE
        STE     RB,DATUS+52     STORE IN OUTPUT ARRAY
*
*
*
*       CALL WRITIT TO BOTH PUT VALUES OUT ONTO TAPE AND TO
*       LIST VALUES FOR SCIENTISTS PERUSAL
*
        CALL    WRITIT,(DATUS,IDATU2)
*
*
*
REST    L       RF,NBLOCK       FETCH NUMBER OF BLOCKS YET TO BE
*                               PROCESSED
        S       RF,=F'1'        SUBTRACT ONE(ONE WAS JUST USED!)
        ST      RF,NBLOCK       STORE BACK IN NBLOCK FOR FUTURE USE
        L       RD,POSSAVE      LOAD PREVIOUS VALUE OF POSSAVE
        AH      RD,DATH+16      ADD THE BLOCKSIZE
        ST      RD,POSSAVE      STORE IN POSSAVE FOR FUTURE USE
*                               IF WHOLE PHYSICAL RECORD WAS NOT
*                               USED, THEN NEXT PASS YOU WILL
*                               ACCESS DATA STARTING AT THAT POSITION
*
        B       TRYAGN          GO TO TRYAGN TO PROCESS NEXT
*                               REGULAR RECORD
*
*
*
*
*
*
*
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
FILEND   L       RJ,=F'1'          END OF FILE - FINISHED
         L       1,ADDAT           RESTORE ADDRESS OF ARGUMENTS IN CALL
         L       RK,0(1)
         ST      RJ,0(RK)
*
*
GOBACK   BC      0,0               NO OPERATION - LIKE CONTINUE IN
*                                  FORTRAN- ENTER HERE WHEN EXECUTION
*                                  ENCOUNTERED NO DIFFICULTIES - CAN
*                                  RE-INITIALIZE BUGS AS A RESULT
*
*
*          REINITIALIZE BUGS
*
         MVI     BUG5,C'0'
         MVI     BUG6,C'0'
         MVI     BUG7,C'0'
         MVI     BUG8,C'0'
         MVI     BUG9,C'0'
         MVI     BUGA,C'0'
         MVI     BUGB,C'0'
         MVI     BUGC,C'0'
*
*
GOBACK1  L       13,4(13)          RETURN PROCEDURE - ENTER HERE
*                                  DIRECTLY,INSTEAD OF THRU GOBACK,
*                                  WHEN DIFFICULTY ENCOUNTERED.
*                                  CAN THEN CHECK 'BUG' VALUES IN
*                                  DUMP
         RETURN  (14,12),T         RESTORE REGISTERS AND RETURN
*
*
*
*
*
*
*          INITIALIZE VARIABLES AND STORAGE HERE
*
*                    LTORG PLACES ALL LITERALS HERE
         LTORG
*
*        FOR DEBUG USE
*
BUG1     DC      C'0'
BUG2     DC      C'0'
```

```
BUG3     DC      C'0'
BUG4     DC      C'0'
BUG5     DC      C'0'
BUG6     DC      C'0'
BUG7     DC      C'0'
BUG8     DC      C'0'
BUG9     DC      C'0'
BUGA     DC      C'0'
BUGB     DC      C'0'
BUGC     DC      C'0'
*
*
*
*                INPUT TAPE DESCRIPTION-
*
TP191A DCB DSORG=PS,MACRF=(GL),DDNAME=TP191A,EODAD=FILEND,DEVD=TA
TP191B DCB DSORG=PS,MACRF=(GL),DDNAME=TP191B,EODAD=FILEND,DEVD=TA
*
*
*                VARIABLES -
*
ADDAT    DC      F'0'          TEMPORARY LOCATION FOR INPUT
*                              CONTENTS OF 1 - ADDRESS OF CALL
*                              SEQUENCE
BACK     DC      F'0'          INDICATOR SHOWING WHETHER REGULAR
*                              DATA WAS ACCESSED YET,(ZERO,NO -
*                              ONE,YES)
CALDAY   DC      F'0'          CALENDAR DAY OR JULIAN
*                              DAY EQUIVALENT OF
*                              MONTH AND DAY STORED HERE.
CIND     DC      F'0'          TEMPORARY STORAGE LOCATION FOR
*                              INDEX REGISTER VALUE
DATA     DC      F'0'          WILL CONTAIN CORE ADDRESS OF TAPE A
*                              DATA
DATB     DC      F'0'          WILL CONTAIN CORE ADDRESS OF TAPE B
*                              DATA
MONTH    DC      F'0'          MONTH OF DATA - NOTE - IT IS
*                              EQUIVALENCED TO DATE FOR EASE
*                              OF STORAGE INTO MONTH,DAY AND YEAR
*                              FOR THIS REASON ALSO. THESE VALUES
*                              ARE OUT OF ALPHABETICAL ORDER
*                              THEY APPEAR WHERE 'DATE' WOULD
*                              HAVE APPEARED
DAY      DC      F'0'
YEAR     DC      F'0'
```

```
*
*       FOR EASE OF STORAGE, WE USE 'ARRAY' DATE FOR
*       MONTH,DAY,AND YEAR, BUT POSITIONS IN STORAGE
*       ARE GIVEN NAMES MONTH,DAY AND YEAR,(SINCE THEY
*       ARE USED IN CALL TO SUBROUTINE CONV). THUS
*       WE MUST EQUIVALENCE THEM.(DATE AND MONTH).
*
DATE     EQU     MONTH
DATLM    DC      F'2464'          LENGTH OF USABLE INPUT DAT 'ARRAY'
*                                 FROM TAPE - ACTUAL PHYSICAL BLOCK
*                                 LENGTH IS 2520 BUT THIS INCLUDES
*                                 FILL ZEROS
DATLPIN  DC      F'0'             TEMPORARY STORAGE POSITION FOR INDEX
*                                 OF LOOP DATLP2
DATLPIND DC      F'0'             TEMPORARY STORAGE FOR
*                                 DISPLACEMENT FOR DATA ACCESS IN
*                                 DATLP2
FINE     DC      F'0'             INDICATOR AS TO WHETHER TAPES
*                                 WERE MOUNTED CORRECTLY(A ON A, AND
*                                 B ON B - 0 MEANS OK WHILE 1 MEANS
*                                 THEY ARE REVERSED
FPT      DC      2F'0'            TEMPORARY STORAGE AREA USED BY MACRO
*                                 FLOAT
FRTM     DC      F'0'             FRAME TIME STORAGE LOCATION
GRPFIL   DC      F'3,3,3,2,2,2'   GIVES NUMBER OF DETECTORS FOR
*                                 EACH FILTER SEGMENT
GRPFILD  DC      F'3,3,3,3,3,3,2,2,2,2,2,2'
*                                 GRPFILD BREAKS THE DETECTORS DOWN
*                                 SO AS TO CORRESPOND TO THE SEE-SAW
*                                 ACCESS OF THE DATA ON THE 2 TAPES.
GVAL     DC      F'0'             TEMPORARY STORAGE AREA
*                                 FOR CONTENTS OF REG. RG
HAD1     DC      F'0'             WILL CONTAIN NUMBER OF NON-ZERO
*                                 RADIANCE VALUES ENCOUNTERED FOR A
*                                 PARTICULAR WAVELENGTH
HAD2     DC      F'0'             DISPLACEMENT VALUE FOR STORING
*                                 INTO ARRAY TEMP
ITIME    DC      2F'0'            TIME STORAGE FOR MONTH,DAY  AND
*                                 YEAR,(IN HALF WORDS),FOR USE BY
*                                 SUBROUTINE FRATIM
LENGTH   DC      F'1440'          LENGTH OF OUTPUT ARRAY IN BYTES
NBLOCK   DC      F'-1'            INITIALIZATION OF NUMBER OF BLOCKS
*                                 LEFT FOR PROCESSING - INDICATES
*                                 NONE WERE DONE AS YET- FIRST ENTRY
NCOND    DC      F'0'             TEMPORARY STORAGE FOR NCOND - CALL
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
*                                       SEQUENCE VARIABLE FURNISHED BY THIS
*                                       SUBROUTINE
NMEAS    DC     F'0'                    WILL CONTAIN NUMBER OF MEASUREMENTS
*                                       FOUND ON TAPE - SUPPLIED ON TAPE
NUMAREA  DC     F'20,17,18,19,19,20,17,18,18,18,17,16'
*                                       NUMBER OF RADIANCE REFERENCES TO BE
*                                       FOUND UPON EACH CHANGE OF TAPE
*                                       REFERENCE - FIRST 20 ON A, 13 ON B,
*                                       ETC.
POINTR   DC     F'124,164,204,124,164,204,244,284,324,244,284,324'
         DC     F'4,44,84,4,44,84,524,564,524,564'
         DC     F'364,404,364,404,444,484,444,484'
*                                       GIVES DISPLACEMENT FROM
*                                       BEGINNING OF DATA BLOCK
*                                       TO RADIANCE DATA IN ORDER
POSAB    DC     F'762,762,802,802,722,722,922,922,842,842'
         DC     F'882,882'
*                                       GIVES DISPLACEMENTS TO
**                                      WAVELENGTH DATA IN ORDER
*                                       ON THE TWO TAPES
*                                       IT SHOULD BE NOTED THAT IN
*                                       THIS AND IN POINTR, ODD
*                                       ENTRIES REFER TO TAPE A
*                                       WHILE EVEN ONES REFER TO
*                                       TAPE B
POSSAVE  DC     F'0'                    INITIALIZATION OF VALUE OF INDEX TO
*                                       ACCESS DAT
PTAN     DC     4F'0'                   TEMPORARY STORAGE FOR
*                                       POINTING ANGLES DERIVED
*                                       IN SUBROUTINE ZENPTN
RLSAVE   DC     F'0'                    TEMPORARY STORAGE FOR REGISTER RL,
*                                       (IT IS USED ELSEWHERE)
SAVE     DC     18F'0'                  SAVE AREA FOR LINKAGE
*
SLIN     DC     F'716,724,732,716,724,732,740,748,756,740'
         DC     F'748,756,692,700,708,692,700,708,796,804'
         DC     F'796,804,780,788,796,804,780,788,796,804'
         DC     F'780,788,796,804,780,788,796,804,780,788'
         DC     F'796,804,780,788,796,804,780,788,796,804'
         DC     F'780,788,796,804,780,788,796,804,780,788'
         DC     F'796,804,780,788,780,788'
*                                       GIVES DISPLACEMENT IN DATH
*                                       TO CALCULATED SLOPE AND
*                                       INTERCEPT VALUES FOR
*                                       RADIANCE DATA IN ORDER
```

```
TEMP     DC      3F'0'           TEMPORARY STORAGE FOR CALCULATED
*                                VALUES OF RADIANCE FOR AVERAGING
*                                PURPOSES
ZENITH   DC      F'0'            TEMPORARY STORAGE FOR VALUE
*                                OF ZENITH ANGLE(SUN) DERIVED
*                                FROM SUBROUTINE ZENPT'
*
*                STORAGE-
*
DATH     DS      360F            OUTPUT ARRAY OF HOUSEKEEPING DATA -
*                                NEEDED FOR PROCESSING OF REGULAR DATA
DATUS    DS      362F            OUTPUT ARRAY OF REGULAR DATA
*                                WHICH WILL BE PUT ONTO NEW
*                                TAPE BY SUBROUTINE WRITIT
*
*       WE ALSO EQUIVALENCE DATUS AND IDATU2 SO THAT WHEN PASSED
*       BY CALL TO SUBROUTINE WRITIT, HALF WORD VALUES MAY BE USED
*
IDATU2   EQU     DATUS
*
*
         END
// EXEC LINKGO,REGION.GO=200K
//*    UNIT 6 IS THE DEBUG AND GENERAL MESSAGE UNIT OR PRINTER
//*     UNIT 9 CONTAINS THE LISTING OF THE NEW TAPE'S CONTENTS
//*       (IT IS DEFINED AS A PRINTER UNIT ALSO)
//*      UNIT 12 IS THE OUTPUT UNIT FOR THE NEW TAPE
//*     TP191A IS THE INPUT DATA TAPE OF FORM'A'
//*     TP191B IS THE INPUT DATA TAPE OF FORM'B'
//GO.FT06F001 DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=7265),
// SPACE=(CYL,(7,3))
//GO.FT09F001 DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=7265),
// SPACE=(CYL,(15,5))
//GO.FT12F001 DD UNIT=(2400-9,,DEFER),DCB=(RECFM=VBS,
// LRECL=1796,BLKSIZE=30536,DEN=3),LABEL=(,BLP),
// VOL=SER=DUMMY
//GO.TP191A DD UNIT=2400-9,DCB=(RECFM=U,BLKSIZE=2520),LABEL=(,BLP),
// VOL=SER=L0003
//GO.TP191B DD UNIT=2400-9,DCB=(RECFM=U,BLKSIZE=2520),LABEL=(,BLP),
// VOL=SER=L0004
//GO.SYSUDUMP DD SYSOUT=A
//GO.DATA5 DD *
L0036
.FALSE.
 -9
```

```
.FALSE.
-S
// EXEC NOTIFY,MODE=ALL,MSG='JOB IS COMPLETE'
/*
// EXEC NOTIFY,MODE=ALL,MSG='LD1 DONE'
/*
/*
/*
/*
/*
/*
/*
/*
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

APPENDIX C - RD192 RUN PROGRAM LISTINGS, INPUT, AND JCL OUTPUT

```
//ZBMJM192 JOB (S10132604D,T,L00360,H01H01),B16,MSGLEVEL=1
//* THISDATE READ AND LIST S-192 TAPES
// EXEC FORTRANH,PARM='OPT=2,EBCDIC,LIST,ID,MAP,XREF',REGION=225K
//SOURCE.SYSIN DD *
C
C         THIS PROGRAM WAS WRITTEN BY M.MACK OF CSC FOR TASK 092,
C         (SKYLAB), UNDER CONTRACT NAS-5-11999 IN FEBRUARY,1974.
C         IT UTILIZES SUBROUTINE RD192 TO READ S-192 TAPES AND, IF
C         DESIRED, PRINT THEIR CONTENTS BEFORE RETURNING CONTROL
C         TO THIS MAIN OR CALLING PROGRAM
C
C         NVAL IS AN INDICATOR ARRAY -
C           1. ITS FIRST ENTRY IS NCOND -
C                 IF 0, IT INDICATES REGULAR DATA WAS PRESENT
C                 IF 1, DATA WAS ERRONEOUS
C                 IF 2, AN END OF FILE WAS ENCOUNTERED
C           2. SECOND ENTRY IS NHOUS -
C                 IF 0, NO HOUSEKEEPING DATA IS PRINTED
C                 IF 1, HOUSEKEEPING DATA IS PRINTED
C           3. THIRD ENTRY IS NSCAN -
C                 IF 0,NO SCAN DATA IS PRINTED
C                 IF 1, ALL SCAN DATA IS PRINTED
C
C         THE REQUESTS OF THE USER CONCERNING PRINTOUT
C         (SETTING OF VARIABLES NHOUS AND NSCAN),ARE
C         TRANSMITTED VIA DATA5 NAMELIST,(ASK). THE
C         LOGICAL INPUTS TO IT ARE NHOUST AND NSCANT.
C         IF SET TO TRUE, THE ASSOCIATED INTEGER IS
C         SET TO 1 AND DATA IS PRINTED. THE DEFAULT
C         VALUE IS FALSE.
C
      LOGICAL NHOUST / .FALSE. /
      LOGICAL NSCANT / .FALSE. /
      DIMENSION SCAN(1093),NVAL(3)
      EQUIVALENCE (NVAL(1),NCOND),(NVAL(2),NHOUS),(NVAL(3),NSCAN)
      DATA NVAL /0,0,0/
      NAMELIST /ASK/ NHOUST,NSCANT
C
      READ(5,ASK)
      IF(NHOUST)NHOUS=1
      IF(NSCANT)NSCAN=1
   10 CONTINUE
      CALL RD192(SCAN,NVAL)
      IF(NCOND.EQ.0) GO TO 10
      IF(NCOND.EQ.1) CALL ERR192(1)
```

```
      IF(NCOND.EQ.2) CALL ERR192(2)
      RETURN
      END
// EXEC ASMG
//SOURCE.SYSIN DD *
// EXEC LINKGO,REGION.GO=220K
//LINK.OBJECT DD *
 INCLUDE LOADLIB(ZBMJMR92)
//* LOADLIB ZBMJMR92 CONTAINS READ PROGRAMS FOR S-192
//*    UNIT 6 IS THE DEBUG AND GENERAL MESSAGE UNIT ON PRINTER
//*      UNIT 9 CONTAINS THE LISTING OF THE TAPE CONTENTS
//*       TP192 IS THE INPUT DATA TAPE
//GO.FT06F001 DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=7265),
// SPACE=(CYL,(7,3))
//GO.FT09F001 DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=7265),
// SPACE=(CYL,(15,5))
//GO.TP192 DD UNIT=2400-9,DCB=(RECFM=U,BLKSIZE=32000),LABEL=(,BLP),
// VOL=SER=L0045
//GO.SYSUDUMP DD SYSOUT=A
//GO.DATA5 DD *
 &ASK
 NHOUST=.TRUE.,NSCANT=.TRUE.,
 &END
// EXEC NOTIFY,MODE=ALL,MSG='JOB IS COMPLETE'
```

```
///78MJMR92 JOB (S1012260AD,T,L00360,001H01),816
//* THISDATE LDLIB 192      CREATION OF LOADLIB 78MJMR92
// EXEC FORTRANH,PARM='OPT=2,LIST,XREF,ID,MAP,EBCDIC'
//SOURCE.SYSIN DD *
      SUBROUTINE ERR192(N)
C         SKYLAB, UNDER CONTRACT NAS-5-11999 IN FEBRUARY, 1974. IT IS
C         UTILIZED BY R92, THE MAIN OR DRIVER PROGRAM, TO PRINT ERROR
C
C         THIS SUBROUTINE WAS WRITTEN BY M.MACK OF CSC FOR TASK 092,
C         OR END OF FILE MESSAGES REQUIRED IN THE READING OF S-192 TAPES.
C
C         FORMATS
C
 2000 FORMAT(1HO,'ERROR ENCOUNTERED IN READING S-192 TAPES - ABEND ',
     1 'CALLED TO FACILITATE LOCATION OF PROBLEM'/)
 2100 FORMAT(1HO, 'END OF FILE ENCOUNTERED UPON READING S-192 ',
     1 'TAPE  -  PROGRAM TERMINATES NORMALLY'/)
C
C
      GO TO (10,20),N
   10 CONTINUE
C
C         ERROR ENCOUNTERED IN READING TAPE - CALL ABEND
C
      WRITE(6,2000)
      CALL ABEND
   20 CONTINUE
C
C       END OF FILE ENCOUNTERED
C
      WRITE(6,2100)
      RETURN
      END
      SUBROUTINE PRINTH(HOUSEK,IHOUS,IHOUS2,LHOUS)
C
C       THIS SUBROUTINE WAS WRITTEN BY M.MACK OF CSC FOR
C       TASK 092,SKYLAB, UNDER CONTRACT NAS-5-11999 IN
C       FEBRUARY,1974. IT IS CALLED BY ASSEMBLY LANGUAGE
C       SUBROUTINE 'RD192' TO PRINT HOUSEKEEPING DATA,(IF
C       DESIRED BY SCIENTISTS).
C       THE CALL SEQUENCE IS CALL PRINTH(HOUSEK,IHOUS,IHOUS2,LHOUS)
C       WHERE HOUSEK IS A REAL*4 ARRAY OF HOUSEKEEPING DATA.IHOUS IS
C       THE INTEGER*4 FORM OF THE SAME ARRAY,IHOUS2 IS THE INTEGER*2
C       FORM OF THE SAME ARRAY AND LHOUS IS THE LOGICAL*1 FORM OF
C       THE SAME ARRAY. CALL SEQUENCE IN THE ASSEMBLY LANGUAGE PROGRAM
```


REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
C      WOULD THEREFORE BE CALL PRINTH(HOUSEK,HOUSEK,HOUSEK,HOUSEK).
C      THE REASON FOR THIS REPETITION IS THAT SOME DATA IS REAL*4,
C      SOME INTEGER*4,SOME INTEGER*2 AND SOME LOGICAL*1. THIS ALLOWS
C      ACCESSING OF ALL THE DATA IN THIS ROUTINE.
C
C      IN ORDER TO ACCESS THAT DATA.
C
C      ALSO, MANY VALUES IN THE ARRAY HAVE MEANINGS EXPLAINED
C      IN THE DESCRIPTION OF TAPE CONTENT SUPPLIED BY HOUSTON.
C      IN ORDER TO ALLEVIATE THE 'PAGE TURNING' THAT WOULD
C      BE REQUIRED TO 'INTERPRET' VALUES, LOGICAL DATA
C      (EBCDIC) ARRAYS ARE SET UP CONTAINING THE APPROPRIATE
C      MESSAGES, AND THESE, NOT THE INDICATORS, WILL BE
C      PRINTED
C
C
       REAL*4 HOUSEK(655),CLSL(4),COLSEL(4,3),PROC(3),PROCES(3,2),
      1TRUNCA(3),IHOUS(655),SCNT(2,2),SCN(2),INPDEV(2,2),IND(2),IMF(4),
      2IMFORM(4,4),CAM(2),CAMR(2,2),TYPEDT(4,2),TYPED(4),CALSRC(2),
      3DATIND(6,2),DATD(6),COLORS(3,4),COLOR(3),TRANS( 9)
       LOGICAL*1 LHOUS(2620)
       INTEGER*2 IHOUS2(1310)
       DATA NOPG /0/
       DATA CALSRC /'LOW ','HIGH'/
       DATA CAMR /'70 M','M   ','5 IN','CH  '/
       DATA COLORS/'NOT ','ACTI','VE  ','RED ','    ','    ','GREE',
      1  'N   ','    ','BLUE','    ','    '/
       DATA COLSEL /'NO C','OLOR','    ','    ','ASSI',
      1'GNED',' COL','OR  ','FALS','E CO','LOR ','    '/
       DATA DATIND /'VIDE','O OR','DERE','D BY',' CHA',
      1 'NNEL','VIDE','O OR','DERE','D BY',' PIX','EL  '/
       DATA IMFORM / 'SING','LE I','MAGE','    ','OVER',
      1 'LAY ','IMAG','ES  ','ABUT',' IMA','GES ','    ',
      2'OFFS','ET I','MAGE','S   '/
       DATA INPDEV /'9 TR','K   ','HDT ','    '/
       DATA PROCES /'SERI','ALLY','    ','CONC','URRE',
      1 'NTLY'/
       DATA SCNT /'LINE','AR  ','SMOO','THED'/
       DATA TRANS /'(1 M','EANS',' ACT','IVE',',',' 0 M',
      1 'EANS',' NOT',' ACT','IVE)'/
       DATA TRUNCA /'LOW ','HIGH','NONE'/
       DATA TYPEDT /'RAW ','DATA','    ','    ','PROC',
      1 'ESSE','D DA','TA  '/
C
C
```

```
C      FORMATS
C
C
 2000 FORMAT(1H1,50X,'HOUSEKEEPING DATA - SKYLAB S-192'/,
     1 1X,62X,'PAGE ',I3/)
 2050 FORMAT(1HO,3X,'COMPUTING SYSTEM ID ',8A4,10X,
     1 'TAPE LIBRARY ID ',5A4,10X,'SENSOR ID ',2A4//,1X,1X,
     2 'DATE OF TAPE GENERATION WAS ',3I3,4X,'TAPE SEQUENCE ID ',
     3 I3,4X,'MISSION NO. ',I3,4X,'SITE ',I3,4X,'LINE ',I4,4X,
     4 'RUN ',I3,4X,'ORBIT ',I4///,1X,37X,'TIME OF FIRST SCAN',
     5 ' IN THIS JOB ',3I3,2X,3I3,2X,I5//)
 2100 FORMAT(1HO,42X,'CHANNELS ACTIVE',1X, 9A4//,1X,34X,16I4/,
     1 1X,34X,16I4/,1X,34X,16I4/,1X,34X,16I4//,1X,9X,
     2 'PROCESSING FLAG ',I2,',I.E. ',4A4,9X,'NO. ACTIVE',
     3 ' CHANNELS ',I3,9X,'NO. BITS IN A PICTURE ELEMENT ',
     4 I2//,1X,8X,'START ADDRESS OF VIDEO DATA WITHIN SCAN ',
     5 I5,15X,'START ADDRESS OF FIRST CALIBRATION AREA WITHIN SCAN ',
     6 I5//,1X,37X,'NO. VIDEO ELEMENTS PER SCAN WITHIN A SINGLE ',
     7 'CHANNEL ',I5/)
 2200 FORMAT(1HO,16X,'NO. OF CALIBRATION ELEMENTS IN THE FIRST ',
     1'CALIBRATION AREA WITHIN THE SCAN IN A SINGLE CHANNEL ',
     2 I5///,1X,'PHYSICAL RECORD SIZE(BYTES) ',I5,6X,
     3 'NO. OF CHANNELS PER PHYSICAL RECORD ',I3,6X,'NO. OF ',
     4 'PHYSICAL RECORDS PER SCAN PER CHANNEL ',I3//,1X,
     5 14X,'NO. OF RECORDS TO MAKE A COMPLETE DATA SET ',I3,
     6 14X,'LENGTH OF THE ANCILLARY BLOCK (BYTES) ',I5//,1X,
     7 10X,'DATA ORDER INDICATOR ',I2,',I.E. ',6A4,10X,
     8 'START PIXEL NO. ',I5,10X,'STOP PIXEL NO. ',I5//,
     9 1X,9X,'OFFSET CONSTANT ',I5,7X,'WORD SIZE OF GENERATING',
     A ' COMPUTER ',I5,7X,'NO. OF DATA SETS PER PHYSICAL RECORD ',
     B I5/)
 2400 FORMAT(1HO,54X,'--CONVERSION FACTORS--'//,1X,'CHANNEL',
     13X,'INTERCEPT COMPONENTS',6X,'INTERCEPT',7X,'SLOPE ',
     2 'COMPONENTS',8X,'SLOPE',11X,'COLOR CODE',3X,
     3 'SCALE INFORMATION'//)
 2500 FORMAT(1H ,2X,I2,'.',2X,5X,I6,4X,I5,6X,E14.7,6X,
     1 I6.2X,I5,5X,E14.7,4X,3A4.11X,I3,7X)
 2600 FORMAT(1HO,54X,'WAVELENGTH OF EACH CHANNEL'/,1X,5X,'CHANNEL',
     1 ' NO.',5X,'STARTING WAVELENGTH',5X,'ENDING WAVELENGTH',
     2 8X,'CHANNEL NO.', 5X,'STARTING WAVELENGTH',5X,
     3 'ENDING WAVELENGTH'//)
 2700 FORMAT(1H ,9X,I3,15X,8A1,15X,8A1,16X,I3,15X,8A1,15X,8A1,4X)
 2800 FORMAT(1HO,'START ADDRESS OF SECOND CALIB. AREA WITHIN ',
     1 'SCAN ',I5,5X,'NO. OF CALIB. ELEMENTS IN SECOND CALIB. ',
     2 'AREA IN SCAN IN ONE CHANNEL ',I5//,1X,51X,
```


REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
     3 'CALIBRATION SOURCE INDICATOR-'/,1X,61X,'FIRST ',A4/,
     4 1X,60X,'SECOND ',A4//,1X,12X,'NO. CHANNELS IN FIRST ',
     5 'PHYSICAL REC. OF DATA SET ',I3,12X,'TOTAL NO. OF ',
     6 'BYTES PER SCAN PER CHANNEL ',I5/)
2900 FORMAT(1H0,6X,'PIXEL SKIP FACTOR ',I3,5X,'SCAN SKIP FACTOR ',
     1 I3,5X,'TOTAL NO. OF SCAN LINES PER FRAME ',I5,5X,
     2'ALTITUDE IN METERS ',I6//,1X,14X,'GROUND SPEED',
     3 '(METERS/SEC.) ',I6,13X,'SCAN TYPE ',2A4,13X,
     4 'ANGLE OF ARC (DEGREES) ',I3//,1X,21X,'CAMERA ',2A4,20X,
     5 'INPUT DEVICE ',2A4,20X,'TRUNCATION ',A4//,1X,56X,
     6 'CHANNELS REQUESTED-'/,1X,18X,32I3/,1X,18X,32I3/)
2950 FORMAT(1H0,'PROCESSING MODE ',3A4//,1X,55X,
     1 'OVERLAY IMAGE FACTORS-'/,1X,32I4/,1X,3314//,1X,7X,
     2 'COLOR SELECT ',2X,4A4,4X,'IMAGE FORMAT ',2X,4A4,4X,
     3 'REPEAT OF PIXELS PER SCAN ',I3,6X,'REPEAT OF SCAN ',
     4 I3//,1X,'PARTIAL SCAN-FROM PIXEL ',I5,' TO PIXEL ',
     5 I5,',(IF ZEROS,TOTAL SCAN)'//,1X,15X,
     6 'SENSOR SCAN RATE IN SCANS/SEC. ',I5,15X,'PIXEL SIZE ',I5,
     7 15X,'ANGLE OF DRIFT ',F6.2/,1X,32X,'ITITLE:',60A1/)
C
C
C       FIRST, INDICES FOR ERCDIC 'EXPLANATIONS' ARE SET UP BASED
C       UPON INDICATOR VALUES FOUND IN HOUSEKEEPING ARRAY.
C
C
C       ID IS THE INDICATOR FOR THE TYPE OF DATA FOUND ON TAPE
C
      ID=IHOUS2(111)+1
      DO 10 L=1,4
   10 TYPED(L)=TYPEDT(L,ID)
C
C       IO IS THE DATA ORDER INDICATOR
C
      IO=IHOUS2(123)+1
      DO 20 L=1,6
   20 DATO(L)=DATIND(L,IO)
C
C       ICAM IS THE INDICATOR FOR THE CAMERA USED
C
      ICAM=IHOUS2(1038)+1
      CAM(1)=CAMR(1,ICAM)
      CAM(2)=CAMR(2,ICAM)
C
C       ISCT IS THE INDICATOR OF SCAN TYPE
C
```

```
      ISCT=IHOUS2(1036)+1
      SCN(1)=SCNT(1,ISCT)
      SCN(2)=SCNT(2,ISCT)
C
C       INP IS THE INDICATOR OF THE INPUT DEVICE
C
      INP=IHOUS2(1038)+1
      IND(1)=INPDEV(1,INP)
      IND(2)=INPDEV(2,INP)
C
C       ITRUNC IS THE INDICATOR OF THE TRUNCATION METHOD
C
      ITRUNC=IHOUS2(1040)+1
      TRUNC=TRUNCA(ITRUNC)
C
C       IPRCS IS THE INDICATOR OF THE PROCESSING METHOD USED
C
      IPRCS=IHOUS2(1105)+1
      DO 30 L=1,3
   30 PROC(L)=PROCES(L,IPRCS)
C
C       ICS IS THE INDICATOR OF THE COLOR SELECTED
C
      ICS=IHOUS2(1171)+1
      DO 40 L=1,4
   40 CLSL(L)=COLSEL(L,ICS)
C
C       IFO IS THE INDICATOR FOR IMAGE INFORMATION
C
      IFO=IHOUS2(1172)+1
      DO 50 L=1,4
   50 INF(L)=IMFORM(L,IFO)
C
C
C       THE CALIBRATION SOURCE INDICATOR WAS FORMED BY
C       ABUTING TWO INDICATOR VALUES IN ONE HALF WORD. HENCE
C       THE HALF WORD CAN HAVE VALUES OF 0,1,256 OR 257.
C       OUR LOW CALIBRATION AND HIGH CALIBRATION SOURCE
C       INDICATORS ARE SET BASED UPON THIS FACT.
C
      IF(IHOUS2(1027) .EQ.     0) CAL1=CALSRC(1)
      IF(IHOUS2(1027) .EQ.     0) CAL2=CALSRC(1)
      IF(IHOUS2(1027) .EQ.     1) CAL1=CALSRC(2)
      IF(IHOUS2(1027) .EQ.     1) CAL2=CALSRC(1)
      IF(IHOUS2(1027) .EQ.   256) CAL1=CALSRC(1)
```


REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

```
      IF(IHOUS2(1022) .EQ.  256) CAL2=CALSRC(2)
      IF(IHOUS2(1027) .EQ.  257) CAL1=CALSRC(2)
      IF(IHOUS2(1027) .EQ.  257) CAL2=CALSRC(2)
C
C       BEGIN LISTING OF HOUSEKEEPING DATA ON NEW PAGE
C
      NOPG=NOPG+1
      WRITE(9,2000) NOPG
      WRITE(9,2050) (HOUSEK(LZ),LZ=1,15),
     1 (IHOUS2(LZ),LZ=31,46)
      WRITE(9,2100) TRANS, (IHOUS2(LZ),LZ=47,111),TYPED,
     1 (IHOUS2(LZ),LZ=112,116)
      WRITE(9,2200) (IHOUS2(LZ),LZ=117,123),DATD,
     1 (IHOUS2(LZ),LZ=124,125),(IHOUS2(LZ),LZ=510,511),
     2 IHOUS2(1024)
C
C       CONTINUE DATA, BUT START ON NEW PAGE
C
      NOPG=NOPG+1
      WRITE(9,2000) NOPG
      WRITE(9,2400)
C
C       SET UP INDICES USED IN LOOP WHICH LISTS SLOPES,
C        INTERCEPTS,COLORS AND SCALE VALUES.
C
C        I1 AND I2 REFERENCE INTERCEPT COMPONENTS
C        I3 AND I4 REFERENCE SLOPE COMPONENTS
C        I5 REFERENCES CALCULATED INTERCEPT VALUES
C        I6 REFERENCES CALCULATED SLOPE VALUES.
C        I7 REFERENCES CORRESPONDING COLOR VALUES
C        I8 REFERENCES CORRESPONDING SCALE VALUES
C
C         THE ORDER OF THE CHANNEL. CHANNELS 14 AND 15 ON THE TAPE
C         ARE NOT ACTIVE. AS THERE ARE 22 ON THE TAPE,THERE ARE 20 ACTIVE
C         THE REASON FOR PRINTING THEM THIS WAY WAS THAT THE
C        NOTE - ALTHOUGH LISTED AS 'CHANNEL NO.', IT IS ACTUALLY
C         'CHANNEL' NO. GIVEN THEM BY HOUSTON DOES NOT CORRESPOND TO
C         THE 'CHANNEL NO. ' BY WHICH THE SCIENTISTS REFERENCE
C         THEM.(ACCORDING TH DRS. CURRAN AND CORB OF GSFC).
C         IT THEREFORE SEEMED USELESS TO RENUMBER THEM HERE BY HOUSTON'S
C         SCHEME.
C         ALSO. ONLY ACTIVE CHANNEL DATA IS LISTED SINCE NON-
C         ACTIVE DATA SHOULD BE ZERO ANYWAY!
C
C
```

```
      I1=122
      I5=604
      I7=380
C        NOACT WILL BE NO. OF ACTIVE CHANNELS
C
      NOACT=IHOUS2(112)
C
      DO 70 LB=1,NOACT
      I1=I1+4
      I2=I1+1
      I3=I1+2
      I4=I3+1
      I5=I5+2
      I6=I5+1
      I7=I7+2
      I8=I7+1
      ICLS=IHOUS2(I7)+1
       DO  60 LT=1,3
   60 COLOR(LT)=COLORS(LT,ICLS)
      WRITE(9,2500) LB,(IHOUS2(LX),LX=I1,I2),HOUSEK(I5),
     1 (IHOUS2(LX),LX=I3,I4),HOUSEK(I6),COLOR,IHOUS2(I8)
   70 CONTINUE
C
C       AGAIN, CONTINUE NEW DATA, BUT ON A NEW PAGE
C
      NOPG=NOPG+1
      WRITE(9,2000) NOPG
      WRITE(9,2600)
C
C       HERE, WAVELENGTH LIMITS FOR EACH 'CHANNEL',(IN
C       ORDER AS GIVEN ON TAPE), ARE LISTED
C
C       LD1 AND LD2 ARE THE INDICES OF THE MINIMUM WAVELENGTH
C       LD3 AND LD4 ARE THE INDICES OF THE MAXIMUM WAVELENGTH
C
      LD1=991
      DO 80 LB1=1,64,2
      LD1=LD1+32
      LD2=LD1+15
      LD3=LD2+1
      LD4=LD3+15
      LB2=LB1+1
      WRITE(9,2700) LB1,(LHOUS(LZ),LZ=LD1,LD2),LB2,
     1 (LHOUS(LZ),LZ=LD3,LD4)
   80 CONTINUE
```

```
C
C        CONTINUE LISTING OF DATA,BUT ON A NEW PAGE
C
      NOPG=NOPG+1
      WRITE(9,2000) NOPG
      WRITE(9,2800) (IHOUS2(LX),LX=1025,1026),CAL1,CAL2,
     1 (IHOUS2(LX),LX=1028,1029)
      WRITE(9,2900) (IHOUS2(LZ),LZ=1030,1032),IHOUS(517),
     1 IHOUS2(1035),SCN,IHOUS2(1037),CAN,IND,TRUNC,
     2 (IHOUS2(LZ),LZ=1041,1104)
      WRITE(9,2950) PROC,(IHOUS2(LX),LX=1106,1170),CLSL,
     1 INF,(IHOUS2(LZ),LZ=1173,1178),HOUSEK(590),
     2 (LHOUS(LZ),LZ=2361,2420)
C
C        RETURN TO CALLING PROGRAM
C
      RETURN
      END
      SUBROUTINE PRNTSC(SCAN,ISCAN,ISCAN2,LSCAN)
C
C        THIS SUBROUTINE WAS WRITTEN BY M.MACK OF CSC FOR TASK 092
C        (SKYLAB) UNDER CONTRACT NAS-5-11999 IN FEBRUARY,1974.
C        IT WILL PRINT THE SCAN ARRAY GENERATED IN READING THE
C        HOUSTON JSC GENERATED S-192 TAPES. THIS SCAN INCLUDES
C        DATA FROM THE ANCILLARY BLOCK AS WELL AS CALCULATED SCAN
C        DATA FOR ONE SCAN LINE FOR ONE CHANNEL.
C        CALL OF THIS PROGRAM BY SUBROUTINE RD192 IS DETERMINED
C        BY USER SUPPLIED ARGUMENTS TO THAT PROGRAM.
C
C        INPUT ARRAYS ARE ACTUALLY ALL THE SAME ARRAY,BUT ARE
C        NECESSITATED BY THE EXISTENCE OF REAL*4,INTEGER*4,
C        INTEGER*2 AND LOGICAL*1 DATA WITHIN THE SAME ARRAY.
C        SINCE EQUIVALENCING IS ILLEGAL WHEN DEALING WITH A
C        PASSED ARRAY, ALL 'NAMES' ARE PASSED. ALSO,
C        VALUES APPEARING IN OTHER THAN REAL*4 FORMS. ALSO,
C        AS PRINTING OF MOST DATA CONTAINED IN THE ANCILLARY
C        BLOCK FOR EVERY CHANNEL FOR EACH LINE WOULD WASTE
C        PAPER,(THIS DATA REMAINS CONSTANT FOR ALL CHANNELS
C        FOR ONE LINE), ONLY PIXEL OR SCAN DATA IS PRINTED
C        FOR ALL BUT THE FIRST CHANNEL.
C
C        ALSO, EACH DATA PAGE PRINTED IS NUMBERED CONSECUTIVELY,
C        BY UTILIZING VARIABLE NOPG,(INITIALIZED TO 0).
C
C        FORMATS
```


REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
C
 2000 FORMAT(1H1,56X,'SKYLAB S-192 DATA'/,1X,61X,'PAGE ',I5//)
 2100 FORMAT(1HO,58X,'-CHANNEL STATUS-'/,21X,2214/,41X,'(ZERO MEANS ',
     1 'ACTIVE WHILE ONE MEANS NOT ACTIVE)'/)
 2200 FORMAT(1HO,'--SPACECRAFT--'/,1X,3X,'LATITUDE ',I3,' DEG. ',
     1 I4,' MIN. ',I4,' OSEC. ',3X,'W. LONGITUDE ',I3,' DEG. ',
     2 I4,' MIN. ',I4,' OSEC. ',3X,'ROLL ',I3,' DEG. ',
     3 I4,' MIN. ',I4,' OSEC. '//,1X,6X,'PITCH ',I3,
     4 ' DEG. ',I4,' MIN. ',I4,' OSEC. ',6X,'YAW ',
     5 I3,' DEG. ',I4,' MIN. ',I4,' OSEC. ',6X,
     6 'ZENITH ',I3,' DEG. ',I4,' MIN. ',I4,' OSEC. ')
 2300 FORMAT(1HO,57X,'--IMAGE FOOTPRINT--'/,1X, 8X,
     1 'CENTER OF FIRST PIXEL HAS ','LATITUDE ',
     2 I3,' DEG. ',I4,' MIN. ',I4,' OSEC. ',
     3 ' AND W. LONGITUDE '.I3,' DEG. ',I4,' MIN. ',
     4 I4,' OSEC. '/,1X, 7X,'CENTER OF CENTER PIXEL HAS',
     5 ' LATITUDE ',I3,' DEG. ',I4,' MIN. ',I4,' OSEC. ',
     6 ' AND W. LONGITUDE ',I3,' DEG. ',I4,' MIN. ',I4,
     7 ' OSEC. '/,1X, 8X,'CENTER OF LAST PIXEL HAS LATITUDE ',
     8 I3,' DEG. ',I4,' MIN. ',I4,' OSEC. ',
     9 'AND W. LONGITUDE ',I3,' DEG. ',I4,' MIN. ',I4,
     A ' OSEC. '/)
 2400 FORMAT(1HO,5X,'HOT BLACKBODY TEMPERATURE SENSOR ''1A'' ',
     1 A4,' TENTH DEG. K ',9X,'COLD BLACKBODY TEMPERATURE '
     2 'SENSOR ''2A'' ',A4,' TENTH DEG. K '/,1X,13X,
     3 'LAMP 1 CURRENT MONITOR ',A4,' MILLIAMPS ',
     4 12X,'REDUNDANT BLACKBODY TEMP. SENSOR ''3A'' ',
     5 A4,' TENTH DEG. K '/,1X,12X,'LAMP 2 CURRENT MONITOR ',
     6 A4,' MILLIAMPS ',11X,'DETECTOR ARRAY TEMP. SENSOR ''A'' ',
     7 A4,' TENTH DEG. K '/,1X,13X,'PRIMARY MIRROR TEMP. SENSOR ',
     8 A4,' TENTH DEG. K ',12X,'SECONDARY MIRROR TEMP. SENSOR ',
     9 A4,' TENTH DEG. K ')
 2500 FORMAT(1HO,12X,'PRIMARY MIRROR TEMP. SENSOR 2 ',A4,
     1 ' TENTH DEG. K ',13X,'ASPHERIC MIRROR TEMP. SENSOR ',A4,
     2 ' TENTH DEG. K '/,1X,4X,'COVER TEMP. SENSOR ',A4,
     3 ' TENTH DEG. K ',4X,'SCAN MOTOR TEMP. SENSOR ',
     4 A4,' TENTH DEG. K ',4X,'COOLER MOTOR TEMP. ',
     5 A4,' TENTH DEG. K '/,16X, 'SCANNER ELECTRONIC TEMP. ',
     6 'SENSOR ',A4,' TENTH DEG. K ',14X,'MONOCHROMATOR',
     7 ' TEMP. ',A4,' TENTH DEG. K '/,46X,'DIGITAL ELECTRONIC',
     8 ' TEMP. ',A4,' TENTH DEG. K ')
 2600 FORMAT(1HO,10X,'DETECTOR ARRAY TEMP. SENSOR ''B'' ',A4,
     1 ' TENTH DEG. K ',11X,'HOT BLACKBODY TEMP. SENSOR ''2B'' ',
     2 A4,' TENTH DEG. K '/,1X,8X,'COLD BLACKBODY TEMP. ',
     3 'SENSOR ''2B'' ',A4,' TENTH DEG. K ',8X,
```

```
     4 'REDUNDANT BLACKBODY TEMP. SENSOR ''3RD'' ',A4,
     5 ' TENTH DEG. K '/,1X,7X,'SCAN LINE RESET TIME ',
     6 A4,' TENTH MICROSECONDS',7X,'SCAN SHIFT DELAY FLAG ',2X,4A4,5X,
     7 'AZIMUTH (HEADING) ',I4,' DEG. ')
2700 FORMAT(1H0,4X,'GROUND VELOCITY IN METERS/SEC. ',I5,3X,
     1 'ALTITUDE IN METERS ',I6,3X,'ORBITAL INCLINATION',
     2 ' ANGLE IN QUARTER ARC SEC. ',I10/)
2800 FORMAT(1H0,'CHANNEL ',I2,' OF A TOTAL OF ',I2,' ACTIVE',
     1 ' CHANNELS'/,1X,22X,'NO. OF VIDEO ELEMENTS/SCAN/CHANNEL ',
     2 I5,30X,'LINE NUMBER ',I6//,16X,'TIME WAS ',3I3,2X,3I3,2X,I5,
     3 '  OR  ',I10,2X,'TENTHS OF MILLISECONDS INTO DAY'/)
2900 FORMAT(1H0,'PIXEL VALUES, (RADIANCE UNITS ARE ',
     1 'WATTS/CM**2/MICRON/STER) -'//)
2950 FORMAT(1H ,I4,'.',2X,E14.7,6X,I4,'.',2X,E14.7,6X,
     1 I4,'.',2X,E14.7,6X,I4,'.',2X,E14.7,6X,I4,'.',2X,
     2 F14.7)
C
C
C
C
      INTEGER*2 ISCAN2(1)
      LOGICAL*1 LSCAN(1)
      DIMENSION SCAN(    1),ISCAN(    1)
      LOGICAL SCSHF(4),SCSHFT(4,3)
      DATA NOPG /0/
      DATA NPRINT/0/
      DATA SCSHFT /'MOVE',' TO ','LEFT',' ',
     1 'NO S','HIFT',' ',' ','MOVE',' TO ',
     2 'RIGH','T  '/
C
C       CHECK WHETHER OR NOT THIS IS THE FIRST CHANNEL - IF
C       IT IS NOT, SKIP PRINTOUT OF ANCILLARY TYPE DATA
C
      IMOD=ISCAN2(2)
      NPRINT=NPRINT+1
      IF(MOD(NPRINT,IMOD).NE.1) GO TO 10
C
C       PRINT OUT ANCILLARY DATA - START ON NEW PAGE
C
      NOPG=NOPG+1
      WRITE(9,2000) NOPG
      WRITE(9,2100) (ISCAN2(LX),LX=15,36)
      WRITE(9,2200) (ISCAN2(LX),LX=37,54)
      WRITE(9,2300) (ISCAN2(LX),LX=55,72)
C
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
C       DETERMINE SCAN SHIFT DELAY FLAG INDEX VALUE
C
        ISC=ISCAN(58)+2
        SCSHF(1)=SCSHFT(1,ISC)
        SCSHF(2)=SCSHFT(2,ISC)
        SCSHF(3)=SCSHFT(3,ISC)
        SCSHF(4)=SCSHFT(4,ISC)
        WRITE(9,2400) (ISCAN(LX),LX=37,44)
        WRITE(9,2500) (ISCAN(LX),LX=45,52)
        WRITE(9,2600) (ISCAN(LX),LX=53,57),SCSHF,ISCAN(59)
        WRITE(9,2700) (ISCAN(LX),LX=60,62)
C
C       PIXEL DATA FOR ONE LINE FOR ONE CHANNEL IS LISTED
C
   10   CONTINUE
        I1=-162
   20   CONTINUE
        NOPG=NOPG+1
        WRITE(9,2000) NOPG
        WRITE(9,2800) (ISCAN2(LX),LX=1,4),(ISCAN2(LX),LX=6,12),ISCAN(7)
        WRITE(9,2900)
        I1=I1+225
        I2=I1+224
C
C
C       NOTE - I1 AND I2 ARE THE LIMITS WITHIN THE PIXEL DATA
C              WHICH WILL BE PRINTED ON ONE PAGE BEFORE STARTING A
C              NEW PAGE.
C
        IF(I1.LT.963) GO TO 30
C
C       LAST OF DATA - LESS THAN 225 VALUES
C
        I1=963
        I2=ISCAN2(3) +62
   30   CONTINUE
        WRITE(9,2950) (LX, SCAN(LX),LX=I1,I2)
C
C       IF THE SECOND VALUE IS NOT THE LAST ONE IN THE ARRAY,(ISCAN2(3)
C        +62),THERE IS STILL MORE SCAN DATA TO BE PRINTED - GO TO 20 TO
C        START A NEW PAGE
C
        IF(I2.LT.(ISCAN2(3)+62)) GO TO 20
C
C         ALL SCAN DATA FOR ONE LINE FOR ONE CHANNEL HAS BEEN
```

```
C       . LISTED - RETURN
C
        RETURN
        END
         SUBROUTINE HRMNSC(ISCAN,ISCAN2)
C
C        THIS SUBROUTINE WAS WRITTEN BY M.MACK OF CSC UNDER CONTRACT
C        NAS-5-11999 IN SEPTEMBER,1974 FOR TASK 062 SKYLAB.IT CONVERTS THE
C        INPUT TENTHS OF MILLISECONDS FROM MIDNIGHT TO HOUR MINUTE AND
C        SECOND PLUSTENTHS OF MILLISECONDS.
C
        INTEGER*2 ISCAN2(1)
        INTEGER*4 ISCAN(1)
        ITIME=ISCAN(7)
        ISCAN2(9)=ITIME/36000000
        ITIME=ITIME-(ISCAN2(9)*36000000)
        ISCAN2(10)=ITIME/600000
        ITIME=ITIME-(ISCAN2(10)*600000)
        ISCAN2(11)=ITIME/10000
        ISCAN2(12)=ITIME-(ISCAN2(11)*10000)
        RETURN
        END
// EXEC ASMG
//SOURCE.SYSIN DD *
*
*       THIS SUBROUTINE WAS WRITTEN BY M.MACK OF CSC FOR TASK 092,
*       SKYLAB,UNDER CONTRACT NAS-5-11999 IN JANUARY-FEBRUARY,1974.
*       IT READS S-192 DATA TAPES SUPPLIED BY HOUSTON JSC AND
*       PASSES VALUES BACK TO THE CALLING PROGRAM FOR FURTHER
*       PROCESSING.
*
*       THE CALL SEQUENCE IS RD192(SCAN,NVAL) WHERE
*         SCAN WILL BE THE ARRAY OF SCAN DATA FOUND,(AFTER
*           CONVERSION OF ITS VALUES TO ENGINEERING UNITS WITHIN
*           THIS ROUTINE AND
*         NVAL IS THE ARRAY OF INDICATORS OF DIMENSION 3 WHOSE
*           FIRST ENTRY IS NCOND - IF 0, DATA WHICH IS RETURNED
*                                        WAS REGULAR(GOOD) DATA
*                                  IF 1, DATA WHICH IS RETURNED
*                                        WAS ERRONEOUS DATA,(AN
*                                        ERROR OF SOME SORT WAS
*                                        ENCOUNTERED)
*                                  IF 2, AN END OF FILE WAS
*                                        ENCOUNTERED
*                                 (VALUE OF NCOND IS SUPPLIED BY RD192)
```


REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

```
*          SECOND ENTRY IS NHOUS -IF 0, NONE OF THE HOUSEKEEPING
*                                 DATA SHOULD BE PRINTED
*                                IF 1, ALL OF THE HOUSEKEEPING DATA
*                                 SHOULD BE PRINTED
*                                (VALUE OF NHOUS IS USER SUPPLIED)
*          THIRD ENTRY IS NSCAN  -IF 0, NONE OF THE SCAN DATA SHOULD
*                                 BE PRINTED
*                                IF 1, ALL OF THE SCAN DATA SHOULD
*                                 BE PRINTED
*                                (VALUE OF NSCAN IS USER SUPPLIED)
*
*
*        THE FOLLOWING MACRO IS USED THROUGHOUT THE PROGRAM. IT
*        CHANGES A FIXED POINT VALUE TO A FLOATING POINT VALUE.
*
         MACRO
&NAME    FLOAT  &RF,&RC
&NAME    ST     &RC,FPT+4
         LPR    &RC,&RC
         O      &RC,=X'46000000'
         ST     &RC,FPT
         LE     &RF,FPT
         AE     &RF,=X'46000000'
         L      &RC,FPT+4
         LTR    &RC,&RC
         BP     *+6
         LNER   &RF,&RF
         MEND
*
RD192    CSECT
         SAVE   (14,12),,*          STANDARD SUBROUTINE SET-UP
*
*        NOTE - IN CASE DIFFERENT REGISTERS ARE DESIRED AS PROGRAM
*        EXPANDS, LITERAL REPRESENTATIONS ARE SET-UP FOR USE IN THE
*        PROGRAM. THESE CAN BE CHANGED.
*        REGISTERS R0,R1,RD ,RE . AND RF  ARE THE SYSTEM REGISTERS,
*        (SET-UP WITH NAMES HERE ONLY FOR CLARITY AND COMPLETENESS).
*
R0       EQU   0                    SYSTEM REGISTER
R1       EQU   1                    SYSTEM REGISTER,(CONTAINS ADDRESS
*                                   OF CALL INITIALLY)
*                                   LATER, CONATINS ADDRESS OF DATA)
R2       EQU   2
R3       EQU   3
R4       EQU   4
```

```
R5          EQU   5                 SOMETIMES USED IN MVC'S TO ADDRESS
*                                   SCAN ARRAY DATA
R6          EQU   6
R7          EQU   7                 OFTEN USED IN MVC'S TO ADDRESS
*                                   HOUSEK(HOUSEKEEPING)ARRAY
R8          EQU   8
R9          EQU   9                 SEE DESCRIPTION OF DATBASE
RA          EQU   10                BASE REGISTER
RB          EQU   11                BASE REGISTER
RC          EQU   12                BASE REGISTER
RD          EQU   13                SYSTEM REGISTER
RE          EQU   14                SYSTEM REGISTER
RF          EQU   15                SYSTEM REGISTER
DATBASE     EQU   9                 USED AS BASE FOR DATA - ALLOWS
*                                   DATA TO BE DIRECTLY LOADED BACK INTO
*                                   CALL AREA.
*
*
*
            BALR  RA,0              CONTINUATION OF STANDARD SET-UP
            USING HERE1,RA,RB,RC
HERE1       L     RB,BASEAD2        THESE ALLOW USE OF SECOND AND
*                                   THIRD REGISTER
            L     RC,BASEAD3
            B     AGAIN
BASEAD2     DC    A(HERE1+4096)
BASEAD3     DC    A(HERE1+8192)
AGAIN       ST    13,SAVE+4
            LA    R2,SAVE
            ST    R2,8(13)
            LR    13,R2
*
*         DEBUG INSERT
*
            MVI   BUG1,C'1'
*
            ST    1,ADCALL          STORE ADDRESS OF CALL SEQ.
            L     R3,ADCALL
            L     R3,4(R3)
            L     R2,4(R3)
            ST    R2,NHOUS          SECURE INDICATOR OF DESIRE TO
*                                   PRINT HOUSEKEEPING DATA
            L     R2,8(R3)
            ST    R2,NSCAN          SECURE INDICATOR OF DESIRE TO
*                                   PRINT SCAN DATA
            L     DATBASE,0(1)
```

```
         USING SCAN,DATBASE          THIS ALLOWS REFERENCE TO ARRAY
*                                    SCAN IN CODING BUT PERMITS
*                                    DATA TO BE PASSED IMMEDIATELY
*     HAS HOUSEKEEPING DATA BEEN READ AS YET?
*       (I.E. HAS SUBROUTINE ALREADY BEEN CALLED ONCE?)
*
         L     R2,USEDIT             INDICATOR - IF ZERO, THIS IS THE
*                                    FIRST ENTRY,(HOUSEKEEPING DATA
*                                    MUST STILL BE READ): IF ONE,
*                                    IT IS A SUBSEQUENT READ AND SO
*                                    CONTROL IS PASSED TO SCAN
*                                    OR REGULAR DATA READ
         C     R2,=F'0'
         BNE   REGULAR
*
*     SINCE THIS IS THE FIRST CALL, TAPE HAS NOT BEEN OPENED
*     AS YET - THAT MUST BE DONE FIRST
*
         OPEN  (TP192,(INPUT))
         GET   TP192                 INSERT ADDRESS OF DATA INTO REG. 1
         ST    R1,ADDAT              STORE ADDRESS OF DATA INTO ADDAT
*                                    FOR FUTURE USE
*
*     DEBUG AID
*
         MVI   BUG2,C'2'
*
*     BEGIN TO READ HOUSEKEEPING DATA
*
         MVC   HOUSEK(60),0(1)       MOVE THE FIRST 60 BYTES,
*                                    (CONTAINING EBCDIC TITLES,ETC.),
*                                    INTO HOUSEK
         SR    R2,R2                 NEXT, INSERT DATE OF TAPE GEN-
         IC    R2,60(1)              ERATION,(NOTE THAT A LOOP IS IM-
         STH   R2,HOUSEK+64          POSSIBLE HERE SINCE JSC ORDERS
         SR    R2,R2                 LAYMAN AND GODDARD DO,
         SR    R2,R2                 TIME AS THE EUROPEANS OR
         IC    R2,61(1)              MILITARY DO(DAY,MONTH AND
         STH   R2,HOUSEK+62          YEAR) RATHER THAN AS THE
         IC    R2,62(1)              (MONTH,DAY,YEAR). NOTE TOO
         STH   R2,HOUSEK+60          THAT THE CURRENT TAPES
*                                    (FEB.'74) FROM JSC DO NOT
*                                    FOLLOW EITHER - THEY LIST YEAR
*                                    DAY AND MONTH - ERROR, BUT ONE
*                                    WITH WHICH WE MUST CONTEND
```


REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

```
         SR    R2,R2
         IC    R2,63(1)
         STH   R2,HOUSEK+66        TAPE SEQ. ID INSERTED INTO
*                                  HOUSEK
*
*     DEBUG AID
*
         MVI   BUG3,C'3'
*
         LA    R7,HOUSEK
         A     R7,=F'68'
         MVC   0(4,R7),64(1)       MISSION NO. AND SITE NO. ARE
*                                  ENTERED
         SR    R2,R2               LINE NO. IS ENTERED
         IC    R2,68(1)
         STH   R2,HOUSEK+72
         SR    R2,R2
         IC    R2,69(1)            RUN NO. IS ENTERED
         STH   R2,HOUSEK+74
         A     R7,=F'8'
         MVC   0(2,R7),70(1)       ORBIT NO. IS ENTERED
*
*     DEBUG AID
*
         MVI   BUG4,C'4'
*
         LH    R2,72(1)            TIME OF SCAN IS
         STH   R2,HOUSEK+90        RECORDED - AGAIN, ORDER
         SR    R2,R2               DIFFERS FROM THAT ON TAPE
         IC    R2,78(1)            JSC GIVE TENTH OF MILLISEC,
         STH   R2,HOUSEK+78        SEC.,MIN.,HOUR,DAY,MONTH,
         SR    R2,R2               YEAR; WE WISH MONTH,DAY,
         IC    R2,77(1)            YEAR,HOUR,MINUTE,SECOND,
         STH   R2,HOUSEK+80        TENTH OF MILLISEC!
         SR    R2,R2
         IC    R2,79(1)
         STH   R2,HOUSEK+82        (YEAR)
         SR    R2,R2
         IC    R2,76(1)
         SR    R2,R2
         IC    R2,75(1)
         STH   R2,HOUSEK+86        (MINUTE)
         STH   R2,HOUSEK+84        (HOUR)
         SR    R2,R2
         IC    R2,74(1)
```

```
          STH   R2,HOUSEK+88        (SECOND)
          LA    R7,HOUSEK
          A     R7,=F'78'           ENTER THESE VALUES(TIME)ALSO
          LA    R2,SCAN             INTO SCAN ARRAY FOR FUTURE USE
          A     R2,=F'10'
          MVC   0(14,R2),0(R7)
*
*       DEBUG AID
*
          MVI   BUG5,C'5'
*
          LA    R3,8                INITIALIZE CHANNEL1 LOOP INDEX
          LA    R8,80               INPUT DATA INDEX,(CHANGES EVERY
*                                   8TH TIME - JSC DATA STORED BIT BY
*                                   BIT HERE)
          LA    R7,92               OUTPUT INDEX FOR HOUSEK ARRAY
CHANNEL1  LA    R5,24               INITIALIZE DISPLACEMENT FOR
*                                   LEFT SHIFT
          LA    R6,7                INITIALIZE DISPLACEMENT FOR
*                                   RIGHT SHIFT
          LA    R4,8                INITIALIZE INNER CHANNEL2 LOOP
*                                   INDEX
CHANNEL2  SR    R2,R2
          IC    R2,0(R8,1)
          SLL   R2,0(R5)            CUT OFF NEEDLESS LEFT BITS
          SRL   R2,0(R5)            RE-POTITION
          SRL   R2,0(R6)            CUT OFF NEEDLESS RIGHT BITS
          STH   R2,HOUSEK(R7)       STORE RESULTING VALUE
          LA    R7,2(R7)            UPDATE HOUSEK INDEX
          LA    R5,1(R5)            UPDATE LEFT SHIFT VALUE
          S     R6,=F'1'            UPDATE RIGHT SHIFT VALUE
          BCT   R4,CHANNEL2         COMPLETE CHANNEL2 LOOP
*
          LA    R8,1(R8)            UPDATE INPUT DATA INDEX
          BCT   R3,CHANNEL1         COMPLETE CHANNEL1 LOOP
*
*       ACTIVE CHANNELS ARE RECORDED
*
*
*       DEBUG AID
*
          MVI   BUG6,C'6'
*
          SR    R2,R2
          IC    R2,88(1)            RECORD PROCESSING FLAG
```


REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
         STH   R2,HOUSEK+220
         SR    R2,R2
         IC    R2,89(1)          RECORD NO. OF ACTIVE CHANNELS
         STH   R2,HOUSEK+222     IN THIS JOB
         STH   R2,SCAN+2         ALSO RECORD IN SCAN ARRAY FOR
*                                FUTURE USE
         SR    R2,R2
         IC    R2,90(1)          RECORD NO. BITS IN A PIC.
         STH   R2,HOUSEK+224
*
         LA    R7,HOUSEK
         A     R7,=F'226'
         MVC   0(10,R7),91(1)      FURTHER VIDEO DATA INFO. IS
*                                RECORDED,(START ADD. OF VID.
*                                DATA,START ADD. OF 1ST CALIB.
*                                     AREA WITHIN SCAN,NO. VIDEO EL.
*                                PER SCAN WITHIN A SINGLE CHANNEL,
*                                NO. CALIB. EL. IN 1ST CALIB.
*                                AREA WITHIN SCAN IN SINGLE CHAN.,
*                                PHYSICAL RECORD SIZE IN BYTES)
         A     R7,=F'4'
         LA    R3,SCAN
         A     R3,=F'4'
         MVC   0(2,R3),0(R7)       RECORD PHYSICAL RECORD SIZE IN
*                                  SCAN ARRAY ALSO
         SR    R2,R2
         IC    R2,101(1)         RECORD NO. OF CHANNELS PER
         STH   R2,HOUSEK+236     PHYSICAL RECORD
         SR    R2,R2
         IC    R2,102(1)         RECORD NO. PHYS. REC. PER
         STH   R2,HOUSEK+238     SCAN PER CHANNEL
         SR    R2,R2
         IC    R2,103(1)         RECORD NO. OF PHYSICAL REC. TO MAKE
         STH   R2,HOUSEK+240     A COMPLETE DATA SET
         A     R7,=F'12'
         MVC   0(2,R7),104(1)      RECORD LENGTH OF ANCILLARY
*                                  BLOCK IN BYTES
*
         SR    R2,R2
         IC    R2,106(1)         RECORD DATA ORDER INDICATOR
         STH   R2,HOUSEK+244
*
         A     R7,=F'4'
         MVC   0(4,R7),107(1)      START AND STOP PIXEL NOS. ARE SAVED
*
```

```
*        SLOPE AND INTERCEPT COMPONENTS LOOP - NOTE THAT
*         UNLIKE THE JSC METHOD OF STORAGE,(ALL A0 VALUES,
*         THEN ALL E0 VALUES,THEN ALL A1 VALUES, THEN ALL E1
*         VALUES), HERE, A0,E0,A1 AND E1 VALUES FOR ONE CHANNEL
*         WILL BE STORED CONTIGUOUSLY.
*
*
*        DEBUG AID
*
          MVI    BUG7,C'7'
*
          LA     R2,92              INITIALIZE ACCESS DISPLACEMENT
*                                   TO INDICATORS IN HOUSEK ARRAY OF
*                                   ACTIVITY(OR NON-ACTIVITY)OF CHAN.
          LA     R3,64              INDEX FOR VALUE1 LOOP
          LA     R8,111             DISPLACEMENT FOR INPUT ARRAY
          LA     R7,HOUSEK            DISPLACEMENT FOR OUTPUT HOUSEK
          A      R7,=F'250'           IS COMPLETELY INITIALIZED
*                                   ARRAY
VALUE1    LH     R5,HOUSEK(R2)        INSERT ACTIVITY INDICATOR FOR
*                                     CHANNEL
          LTR    R5,R5                TEST IF ACTIVE
          BZ     COMLOOP              IF NOT, SKIP INFO.(ALL ZERO)
          LA     R6,4                 INDEX FOR INNER VALUE2 LOOP
          SR     R4,R4              DISPLACEMENT NEEDED FOR SECURING
*                                   A0,E0,A1,E1 VALUES CONTIGUOUSLY
VALUE2    LR     R5,R8
          AR     R5,R4              TOTAL DISPLACEMENT TO ELEMENT
          AR     R5,1               ADD BEGINNING ADDRESS OF DATA
          MVC    0(2,R7),0(R5)
          LA     R4,128(R4)         UPDATE COMPONENT DISPLACEMENT
          LA     R7,2(R7)           UPDATE OUTPUT ARRAY DISPLACEMENT
          BCT    R6,VALUE2          COMPLETE INNER VALUE2 LOOP
*
COMLOOP   LA     R8,2(R8)             UPDATE INPUT INDEX DISPLACEMENT
*                                   TO REFLECT CHANGE IN CHANNEL
          LA     R2,2(R2)             UPDATE INDEX TO INDICATORS OF
*                                     ACTIVITY OR NON-ACTIVITY
          BCT    R3,VALUE1          COMPLETE OUTER VALUE1 LOOP
*
*
          MVI    BUG8,C'8'
*
*        DEBUG AID
*       COLOR CODE AND SCALE FACTOR LOOP - AGAIN, VALUES, HERE,
```


REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
*       WILL BE STORED CONTIGUOUSLY FOR EACH CHANNEL
*
          LA     R2,92              INITIALIZE IDSPL. TO INDICATOR OF
*                                   ACTIVITY OR NON-ACTIVITY OF CHAN.
          LA     R3,64              INDEX OF OUTER COLSCL1 LOOP SET
          LA     R8,623             DISPLACEMENT FOR INPUT ARRAY SET
          LA     R7,762             DISPLACEMENT FOR OUTPUT ARRAY,
*                                   HOUSEK,SET
COLSCL1   LH     R5,HOUSEK(R2)        INSERT INDICATOR OF ACTIVITY OF
*                                     CHANNELS
          ST     R2,CIND              STORE R2 TEMPORARILY SO FREE IT
*                                     FOR POSSIBLE USE BELOW
          LTR    R5,R5                TEST IF CHANNEL ACTIVE
          BZ     COMCOL               IF NOT ACTIVE, DO NOT RECORD INF.
          LA     R6,2                 INDEX FOR INNER COLSCL2 LOOP SET
          SR     R4,R4              DISPLACEMENT NEEDED FOR SECURING
*                                   COLOR CODE AND SCALE FACTOR VALUES
*                                   CONTIGUOUSLY
*
COLSCL2   LR     R5,R8
          AR     R5,R4              TOTAL DISPLACEMENT TO ELEMENT
          SR     R2,R2
          IC     R2,0(R5,1)
          STH    R2,HOUSEK(R7)
          LA     R7,2(R7)           UPDATE OUTPUT ARRAY DISPLACEMENT
          LA     R4,64(R4)          UPDATE DISPLACEMENT TO ELEMENT
          BCT    R6,COLSCL2         COMPLETE INNER COLSCL2 LOOP
*
COMCOL    L      R2,CIND              DETERMINE CURRENT VALUE OF DISPL.
*                                     TO INDICATOR ARRAY AND UPDATE IT
          LA     R2,2(R2)
          LA     R8,1(R8)             UPDATEIDISPLACEMENT TO INPUT ARRAY
          BCT    R3,COLSCL1         COMPLETE OUTER COLSCL1 LOOP
*
*       DEBUG AID
*
          MVI    BUG9,C'9'
*
          SR     R2,R2
          IC     R2,751(1)          INSERT OFFSET CONSTANT
          STH    R2,HOUSEK+1018
          SR     R2,R2
          IC     R2,752(1)          INSERT WORDSIZE OF GENERATING
          STH    R2,HOUSEK+1020     COMPUTER
          LA     R7,HOUSEK
```

```
         A      R7,=F'1022'
         LR     R6,1                  INITIALIZE DISPL. TO INPUT DATA
         A      R6,=F'753'
         LA     R3,4                  INITIALIZE INDEX OF MOVEIT LOOP
*
MOVEIT   MVC    0(256,R7),0(R6)       INSERT BEGINNING AND ENDING
*                                     WAVELENGTH LIMITS FOR EACH CHANNEL
         LA     R6,256(R6)            UPDATE DISPL. TO INPUT ARRAY
         LA     R7,256(R7)            UPDATE DISPL. TO OUTPUT ARRAY
         BCT    R3,MOVEIT             COMPLETE MOVEIT LOOP
         SR     R2,R2
         IC     R2,1777(1)          INSERT NO. OF DATA SETS PER
         STH    R2,HOUSEK+2046      PHYSICAL RECORD
*
         A      R7,=F'2'
         MVC    0(4,R7),1778(1)       INSERT START ADD. OF 2ND CALIB.
*                                     AREA WITHIN SCAN AND NO. OF CALIB.
*                                     EL. IN 2ND CALIB. AREA IN SCAN IN
*                                     ONE CHANNEL
         SR     R3,R3
         IC     R3,1782(1)          INSERTING CALIB. SOURCE
         SRL    R3,1                INDICATOR FOR 1ST AND 2ND
         SLL    R3,4                CALIB. AREAS. SHIFTS
         STH    R3,HOUSEK+2052      NECESSITATED BY METHOD OF
         SR     R3,R3               STORAGE
         IC     R3,1782(1)
         SLL    R3,7
         SRL    R3,7
         STH    R3,WORD
         A      R7,=F'5'
         LA     R6,WORD
         A      R6,=F'1'
         MVC    0(1,R7),0(R6)
         A      R7,=F'1'
         MVC    0(8,R7),1784(1)       INSERT NO. CHANNELS IN 1 PHYS.
*                                     REC. OF DATA SET,TOTAL NO.
*                                     BYTES/SCAN/CHANNEL,PIXEL
*                                     SKIP FACTOR AND SCAN SKIP FACTOR
*
*        DEBUG AID
*
           MVI    BUGA,C'A'
*
*        INPUT ARRAY OF HOUSEKEEPING DATA CONTAINS DATA NOT
*         AT ALL USEFUL TO GODDARD PERSONNEL IN BYTES 1793 THRU
```


REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

```
*        2758, SO THESE BYTES ARE NOT ACCESSED HERE.
*
         SR    R3,R3
         ST    R3,WORD              REINITIALIZE WORD FOR LATER USE
         SR    R2,R2
         IC    R3,2756(1)           TOTAL NO OF SCAN LINES PER
         M     R2,=F'1000'          FRAME ARE ACCESSED,(HOUSTON
         STH   R3,HOUSEK+2062       STORED THEM IN N THOUSANDS SO
*                                   VALUE MUST BE MULTIPLIED BY
*                                   1000 TO YIELD CORRECT VALUE)
         ST    R3,TOTLINES          VALUE ALSO STORED IN TOTLINES FOR
*                                   FUTURE USE
         STH   R3,SCAN+8            SIMILARLY, IT IS STORED IN THE
*                                   ARRAY SCAN.
*
*        DEBUG AID
*
         MVI   BUGB,C'B'
*
*
*        GENERAL INFORMATION FOLLOWS - MAY NOT BE NEEDED BUT
*         SAVED ON SLIGHT CHANCE OF USE
*
         LA    R5,WORD
         A     R5,=F'1'
         MVC   0(3,R5),2789(1)
         L     R2,WORD              STORAGE METHOD AGAIN NECESSITATES
         ST    R2,HOUSEK+2064       PECULIAR ACCESS AND INSERTION OF
*                                   ALTITUDE,(HERE,JSC USED 3 BYTES FOR
*                                     IT)
         LH    R2,2792(1)
         STH   R2,HOUSEK+2068       INSERT GROUND SPEED IN M/SEC.
*
*        SCAN INFORMATION LOOP
*         SCAN TYPE,ANGLE OF ARC,CAMERA TYPE,INPUT DEVICE,TRUNCATION
*
         LA    R3,5
         LA    R7,2070              OUTPUT ARRAY DISPLACEMENT
         LA    R8,2794              INPUT ARRAY DISPLACEMENT
SCANINF  SR    R2,R2
         IC    R2,0(R8,1)
         STH   R2,HOUSEK(R7)
         LA    R7,2(R7)             UPDATE DISPLACEMENT FOR OUTPUT ARRAY
         LA    R8,1(R8)             UPDATE DISPLACEMENT FOR INPUT ARRAY
         BCT   R3,SCANINF           COMPLETE SCANINF LOOP
```

```
*
*       CHANNELS REQUESTED LOOP
*
          LA     R5,8                NOTE THAT OUTER LOOP INITIALIZED TO
*                                    8 SINCE 64 CHANNELS' INFORMATION ARE
*                                    STORED BIT BY BIT
          LA     R7,2080             OUTPUT ARRAY DISPLACEMENT SET
          LA     R8,2799             INPUT ARRAY DISPLACEMENT SET
CHNRO1    LA     R6,8                INDEX OF INNER CHNRO2 LOOP - CORRES-
*                                    PONDS TO FACT THAT THERE ARE 8 BITS
*                                    IN A BYTE
          SR     R4,R4               LEFT SHIFT INDEX INITIALIZED
          LA     R3,7                RIGHT SHIFT INDEX INITIALIZED
*
CHNRO2    SR     R2,R2
          IC     R2,0(R8,1)          DATA FROM TAPE INSERTED INTO REGISTER
          STH    R4,CH1+2             ALTER INSTRUCTIONS FOR SHIFT
          STH    R4,CH2+2
          STH    R3,CH3+2
CH1       SLL    R2,R4               CUT OFF DESIRED NO. BITS ON LEFT
CH2       SRL    R2,R4               RESTORE TO ORIGINAL POSITION
CH3       SRL    R2,R3               CUT OFF DESIRED NO. BITS ON RIGHT
          STH    R2,HOUSEK(R7)       STORE IN OUTPUT HOUSEK ARRAY
          LA     R7,2(R7)            UPDATE OUTPUT ARRAY DISPLACEMENT
          LA     R4,1(R4)            UPDATE LEFT SHIFT VALUE
          S      R3,=F'1'            UPDATE RIGHT SHIFT VALUE
          BCT    R6,CHNRO2           COMPLETE CHNRO2 LOOP
*
          LA     R8,1(R8)            UPDATE INPUT ARRAY DISPLACEMENT
          BCT    R5,CHNRO1           COMPLETE CHNRO1 LOOP
*
*       THE NEXT LOOP STORES VARIOUS PARAMETERS INCLUDING:
*        PROCESSING MODE INDICATOR,OVERLAY IMAGE FACTORS,COLOR SELECT
*        INDICATOR,IMAGE FORMAT INDICATOR,REPEAT OF PIXELS PER SCAN,
*        AND REPEAT OF SCAN
*
          LA     R5,70               INITIALIZE INDEX OF REST1 LOOP
          LA     R7,2208             INITIALIZE DISPLACEMENT FOR
*                                    OUTPUT HOUSEK ARRAY
          LA     R8,2807             INITIALIZE DISPLACEMENT FOR INPUT
*                                    ARRAY
*
REST1     SR     R2,R2
          IC     R2,0(R8,1)          INSERT INPUT DATA INTO REGISTER
          IC     R4,0(R8,1)
```


REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
         SLL   R4,24
         LTR   R4,R4
         BP    POSR                SINCE HOUSTON JSC FORES NEGATIVES
*                                  USING LEADING BIT SETTINGS ONLY,
*                                  WE MUST CHECK THAT BIT,WE-SET
*                                  IT TO ZERO IF IT IS ONE, AND,ALSO
*                                  IF IT HAD BEEN ONE, LOAD THE
*                                  NEGATIVE OF THE VALUE
         N     R2,=X'00007FFF'
         LNR   R2,R2
POSR     STH   R2,HOUSEK(R7)       NOW, STORE RESULT IN OUTPUT HOUSEK
*                                  ARRAY
         LA    R8,1(R8)            UPDATE DISPLACEMENT TO INPUT ARRAY
*                                  ARRAY HOUSEK
         BCT   R5,REST1            COMPLETE REST1 LOOP
*
         LA    R7,2(R7)            UPDATE DISPLACEMENT TO OUTPUT
         LA    R7,HOUSEK
         A     R7,=F'2348'
         MVC   0(6,R7),2877(1)       INSERT PARTIAL SCAN INDICATOR AND
*                                  SENSOR SCAN RATE IN SCANS/SEC.
         SR    R2,R2
         IC    R2,2883(1)          INSERT PIXEL SIZE INTO HOUSEK
         STH   R2,HOUSEK+2354
         SR    R2,R2
         IC    R2,2884(1)
         IC    R4,2884(1)
         SLL   R4,24
         LTR   R4,R4
         BP    POSAN
         N     R2,=X'00007FFF'
         LNR   R2,R2
POSAN    BC    0,0                  CHANGE INPUT FIXED POINT VALUE
         FLOAT R2,R2               OF ANGLE OF DRIFT TO FLOATING POINT
         SR    R4,R4
         IC    R4,2885(1)          CHANGE FIXED POINT FRACTION OF
         FLOAT R4,R4               ANGLE OF DRIFT TO FLOATING POINT
         DE    R4,=E'256.'         DIVIDE BY PROPER POWER OF 16
         AER   R2,R4               ADD FRACTION TO WHOLE NO.
         STE   R2,HOUSEK+2356      STORE IN HOUSEK ARRAY
*
         A     R7,=F'12'
         MVC   0(60,R7),2940(1)      INSERT TITLE SUPPLIED BY HOUSTON
*                                  JSC INTO HOUSEK
*
```

```
*
*       DEBUG AID
*
          MVI    BUGC,C'C'
*
*
*       THE NEXT LOOP CALCULATES THE SLOPE AND INTERCEPT VALUES
*        USING INPUT COMPONENTS AND STORES RESULTS IN HOUSEK
*
          LH     R3,HOUSEK+222        LOAD NUMBER OF CHANNELS WHICH ARE
*                                     ACTIVE AS INDEX FOR OUTER SLIN2 LOOP
          LA     R8,250               INITIALIZE DISPLACEMENT TO HOUSEK
*                                     FOR ACCESSING SLOPE AND INTERCEPT
*                                     COMPONENTS
          LA     R7,2420              INITIALIZE DISPLACEMENT TO HOUSEK
*                                     FOR INSERTION OF CALCULATED
*                                     SLOPE AND INTERCEPT VALUES
*
SLIN2     SR     R6,R6                INITIALIZE DISPLACEMENT INDEX USED
*                                     FOR ACCESSING EITHER INTERCEPT COM
*                                     PONENT OR SLOPE COMPONENTS
          LA     R5,2                 INNER SLIN1 INDEX INITIALIZED
*
SLIN1     LR     R4,R6
          AR     R4,R8                FIND TOTAL ADDRESS OF COMPONENT
          ST     R5,CIND              STORE VALUE OF R5 TEMPORARILY IN
*                                     CIND SO R5 AVAILABLE FOR USE
          LH     R2,HOUSEK(R4)
          LTR    R2,R2
          BP     OK                   THE JSC NEGATIVES ARE FORMED WITH
*                                     LEADING BIT ONE; IBM USES 2'S
*                                     COMPLEMENT; VALUE WAS NOT PREVIOUSLY
*                                     CHECKED OR CHANGED SO IT MUST BE
*                                     ACTED UPON HERE.
          N      R2,=X'00007FFF'
          LNR    R2,R2
          STH    R2,HOUSEK(R4)        STORE OUT PROPER VALUE OF COMPONENT
OK        FLOAT  R2,R2                CHANGE VALUE TO FLOATING POINT
          LE     R4,=E'10.'           PREPARE FOR EXPONENT CALCULATION
          LH     R5,HOUSEK+2(R4)      LOAD IN EXPONENT
          LTR    R5,R5                TEST REGISTER TO DETERMINE IF
          BZ     ZERO                 IT IS ZERO? OR
          BM     NEG                  IT IS NEGATIVE?
*
          S      R5,=F'1'             IT IS POSITIVE! CHECK IF IT IS
```


REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
*                                       GREATER THAN ONE,(IF ONE, VALUE
*                                       OF 10 IS ALREADY IN REG. R4)
         BZ      ONE
EXP      ME      R4,=E'10.'             THIS LOOP CALCULATES POWER OF TEN
         BCT     R5,EXP
         B       ONE
NEG      N       R5,=X'00007FFF'        AGAIN, JSC NEGATIVES DIFFER FROM IBM
         LNR     R5,R5
         STH     R5,HOUSEK+2(R4)        STORE CORRECTED COMPONENT BACK
         LPR     R5,R5                  TO DETERMINE POWER OF 10, THE
*                                       POSITIVE VALUE OF EXPONENT MUST
*                                       BE DERIVED; THEN RECIPROCAL
*                                       OF RESULT IS CALCULATED
         S       R5,=F'1'               AS WITH POSITIVE VALUE OF COMPONENT,
*                                       WE INSURE THAT VALUE IS NOT ONE
         BZ      MONE                   IF MINUS ONE, CONTROL PASSES TO
*                                       MONE
EXPN     ME      R4,=E'10.'             THIS LOOP, LIKE ITS POSITIVE
*                                       COUNTERPART EXP, CALCULATES
*                                       THE POWER OF TEN
         BCT     R5,EXPN
*
MONE     STE     R4,FPT                 TO DETERMINE RECIPROCAL, VALUE
*                                       IS TEMPORARILY STORED IN FPT
         LE      R4,=E'1.'
         DE      R4,FPT                 CALCULATE RECIPROCAL
         B       ONE
ZERO     LE      R4,=E'1.'              IF EXPONENT IS ZERO,(10)**0=1.
ONE      MER     R2,R4                  MULTIPLY POWER OF TEN BY
*                                       COEFFICIENT
         STE     R2,HOUSEK(R7)          STORE OUTPUT VALUE INTO HOUSEK
         LA      R7,4(R7)               UPDATE DISPLACEMENT FOR CALCULATED
*                                       SLOPE AND INTERCEPT VALUE
         LA      R6,4                   UPDATE DISPLACEMENT FOR SECOND
*                                       (SLOPE) COMPONENT ACCESS
         L       R5,CIND                RESTORE VALUE OF INDEX TO LOOP
         BCT     R5,SLIN1               COMPLETE SLIN1 LOOP
*
         LA      R8,8(R8)               UPDATE DISPLACEMENT TO COMPONENTS
*                                       PER CHANNEL
         BCT     R3,SLIN2               COMPLETE SLIN2 LOOP
*
*
*        DEBUG AID
*
```

```
          MVI    BUGD,C'D'
*
* ********************END OF HOUSEKEEPING ACCESS************************
*
*       IF DESIRED BY USER, HOUSEKEEPING DATA IS PRINTED USING PRINTH
*
          L      R2,NHOUS              CHECK USER OPTION SUPPLIED IN CALL
          LTR    R2,R2
          BZ     NOTWSHHS
          CALL   PRINTH,(HOUSEK,HOUSEK,HOUSEK,HOUSEK),VL
*
NOTWSHHS  L      R2,=F'1'              INSERT INDICATOR TO SHOW HOUSE-
          ST     R2,USEDIT             KEEPING DATA HAS BEEN ACCESSED
          SR     R2,R2
          ST     R2,CHANNELO           INITIALIZE CHANNEL ORDER VARIABLE
*                                      TO ZERO
          ST     R2,NOFIL              INITIALIZE FILE ORDER NO. TO
*                                      ZERO
          LA     R2,0                  INITIALIZE LINE ORDER NO. TO ZERO
          ST     R2,LINENO
          LA     R2,1
          STH    R2,SCAN+6             STORE LINE NO. ALSO IN SCAN ARRAY
*
*
* ************************************************************************
*
*          END OF ALL HOUSEKEEPING STORAGE,PRINTING ETC.
*
*              REGULAR DATA ACCESS BEGINS
*
*
* ************************************************************************
*
REGULAR   BC     0,0                   'CONTINUE' TYPE STATEMENT INSERTED
*                                      TO ALLOW LOGICAL INSERTION OF DEBUG
*                                      VALUES WHICH MAY LATER BE REMOVED
*
*       DEBUG AID
*
          MVI    BUGE,C'E'
*
          L      R2,NOCHANPR           INITIALLY, THIS VALUE IS ZERO:LATER
*                                      IT WILL GIVE THE NUMBER OF CHANNEL
*                                      (SCAN LINE) RECORDS LEFT IN THE
*                                      PHYSICAL RECORD
```


REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

```
         LTR   R2,R2
         BP    HAVEDATA          IF ZERO, TAPE MUST BE ACCESSED TO
*                                GET A NEW PHYSICAL RECORD: IF NON-
*                                ZERO, DATA IS STILL AVAILABLE FOR
*                                USE FROM THE LAST ACCESS
*
NEEDDATA GET   TP192              INSERT ADDRESS OF DATA INTO REG. 1
         ST    1,ADDAT           STORE ADDRESS OF DATA INTO ADDAT
*                                SUBSEQUENT CALLS WOULD NEED IT!
*
*        DEBUG AID
*
         MVI   BUGF,C'F'
*
         LH    R2,HOUSEK+236     RE-INITIALIZE NO. OF CHANNELS
*                                LEFT IN PHYSICAL RECORD TO
*                                NO. PROMISED IN HOUSEKEEPING DATA
         ST    R2,NOCHANPR
         L     R2,NOFIL          ACCESS CURRENT NO. OF FILES
         LA    R2,1(R2)          ADD ONE - READING A NEW PHYSICAL
*                                RECORD(FILE)
         LH    R3,HOUSEK+240     COMPARE INCREMENTED NO. WITH
         CR    R2,R3             PROMISED MAXIMUM NO. OF PHYSICAL
*                                REC. PER 'DATA REC.'
         BNH   FILENOOK          IF LESS THAN OR EQUAL TO THAT NO.,
*                                THE FILE NO. IS ACCEPTABLE
         LA    R2,1              IF NOT, THIS MUST BE THE START OF
*                                A NEW 'DATA SET'
FILENOOK LH    R3,0(1)           COMPARE FILE NO. ON TAPE SUPPLIED
         CR    R2,R3             BY HOUSTON JSC TO NO. DYNAMICALLY
         BNE   ERRFILNO          CALCULATED;IF UNEQUAL, ERROR!
*
*        DEBUG AID
*
         MVI   BUGG,C'G'
*
         LA    R6,2              INITIALIZE POSITION IN FILE COUNTER
         ST    R6,POSINFIL
         ST    R2,NOFIL          STORE CURRENT VALUE OF FILE NO.
*
HAVEDATA L     R2,CHANNELO       ACCESS CURRENT VALUE OF THE ORDER
*                                OF THE CHANNEL
         LA    R2,1(R2)          INCREMENT IT,(NEW CHANNEL IS NOW
*                                BEING READ
*
```

```
*        DEBUG AID
*
         MVI   BUGH,C'H'
*
         ST    R2,CHANNELO        STORE CURRENT VALUE IN CHANNELO FOR
*                                 LATER ACCESS
         LA    R3,1               IF,AT THIS POINT, CHANNEL ORDER IS 1
         CR    R2,R3              THEN LINE NUMBER MUST BE UPDATED
         BE    CONTIN
         LH    R3,HOUSEK+222      COMPARE ORDER OF CHANNEL TO NO. OF
         CR    R2,R3              ACTIVE CHANNELS PROMISED
         BNH   CHANOK
         LA    R2,1               IF NO. WAS GREATER, THIS MUST BE
         LH    R3,HOUSEK+236      THE BEGINNING OF A NEW GROUP OF
         C     R3,NOCHANPR        CHANNELS. DATA MUST BE ACCESSED IF
         BE    CONTIN             AT THIS POINT, NOCHANPR IS NOT THE
         LA    R2,0               NUMBER 'PROMISED' IN HOUSEKEEPING
         ST    R2,CHANNELO
         B     NEEDDATA
CONTIN   ST    R2,CHANNELO
         L     R2,LINENO          ACCESS LINE NO. - IF BACK TO FIRST
*                                 CHANNEL, LINE NO. HAS CHANGED
         LA    R2,1(R2)           UPDATE LINE NO.
         ST    R2,LINENO          STORE CURRENT VALUE IN LINENO
*                                 AT THIS POINT IN THE ORIGINAL VERSION
*                                 THE LINE NUMBER WAS COMPARED TO THE
*                                 TOTAL NUMBER OF LINES LISTED IN
*                                 HOUSEKEEPING. HOWEVER, HOUSTON JSC
*                                 NOW ASSURES GSFC(OCT.'74),THAT THAT
*                                 VALUE IS INVALID.SO.NO CHECKS MADE.
         STH   R2,SCAN+6          STORE CORRECT LINE NO. INTO SCAN
*                                 ARRAY FOR USER USE
*
*        DEBUG AID
*
         MVI   BUGI,C'I'
*
CHANOK   L     R3,CHANNELO        INSERT CURRENT VALUE OF CHANNEL
*                                 ORDER NO.
         S     R3,=F'1'           DECREMENT IT
         SR    R2,R2              PREPARE FOR MULTIPLICATION - THE
         M     R2,=F'4'           RESULT WILL BE THE DISPLACEMENT
*                                 VALUE NEEDED TO ACCESS CORRECT
*                                 CHANNEL NUMBER
         L     R4,CHNVL(R3)
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
          ST    R4,CHANNEL          STORE CORRECT NO.,(ACTUALLY ONLY
*                                   'CORRECT' SO FAR  AS WHAT IS
*                                   RECORDED BY HOUSTON FOR THAT
*                                   CHANNEL NAME - ACTUAL CHANNEL
*                                   NUMBERS DIFFER-CHANNELS HERE,IN
*                                   PAIRS, ARE COMPLETE INPUT CHANNELS)
          STH   R4,SCAN             STORE CORRECT CHANNEL NO. IN SCAN
*                                   ARRAY
*
*       DEBUG AID
*
          MVI   BUGK,C'K'
*
          SR    R2,R2
          M     R2,=F'2'            MULTIPLY DISPLACEMENT TO CHANNEL
*                                   VALUE ARRAY BY 2 TO GET PROPER
*                                   DISPLACEMENT FOR SLOPE AND
*                                   INTERCEPT ACCESSING,(THERE ARE
*                                   2 WORDS IN THAT ARRAY FOR EACH
*                                   CHANNEL
          ST    R3,SLDISPL          STORE VALUE FOR LATER USE
          LTR   R3,R3
          BP    LATERFIL            IF NON-ZERO,(POSITIVE), CHANNEL
*                                   IS NOT THE FIRST SO ANCILLARY
*                                   BLOCK IS NOT PRESENT,(AND NEED
*                                   NOT BE READ THEREFORE)
*
*       ANCILLARY BLOCK IS READ AT THIS POINT
*         NOTE - DESCRIPTION OF ANCILLARY BLOCK GIVEN BY JSC IN
*         PHO-TR 543 REFERENCES THE POSITION OF VALUES IN THE
*         ANCILLARY BLOCK ITSELF - HOWEVER, THE ANCILLARY BLOCK
*         IS ACTUALLY PRECEDED IN THE FILE BY A 2 BYTE FILE NO.
*         THIS IS EXPLAINED ELSEWHERE IN PHO-TR 543.
*
ANCILRY   L     R8,POSINFIL         INSERT CURRENT VALUE OF
*                                   DISPLACEMENT WITHIN FILE
          LA    R6,SCAN             MOVE DIRECTLY, THE TIME IN TENTHS
          A     R6,=F'24'           OF MILLISEC. AT THE START OF DATA
          LR    R5,1                INITIALIZE R5 TO ADDRESS OF INPUT ED
*                                   PHYSICAL RECORD
          AR    R5,R8               VALUE OF POSINFIL,IN REG.R8,IS ADDED
          MVC   0(4,R6),0(R5)       NOTE THAT AT THIS POINT, POSINFIL
*                                   TWO FROM CODING ABOVE,(AND ASSUMING
*                                   THE DESCRIPTION OF TAPE CONTENTS
*                                   SUPPLIED BY HOUSTON IS CORRECT.
```

```
          LA     R8,4(R8)              UPDATE DISPLACEMENT
          CALL   HRMNSC,((R9),(R9))    UPDATE HOUR,MINUTE AND SECOND VALUES
          L      1,ADDAT               RESTORE ADDRESS OF DATA
*
*       DEBUG AID
*
          MVI    BUGL,C'L'
*
*       RECORD CHANNEL STATUS OF ONLY THE ACTIVE CHANNELS
*
          LA     R4,0                  DISPLACEMENT TO CHNVL ARRAY
          LA     R5,22                  INITIALIZE STATUS LOOP INDEX
          LA     R6,28                  INITIALIZE DISPLACEMENT TO
          LA     R3,1                   POSITION IN SCAN ARRAY FOR
*                                       STATUS VALUES
CHANSTAT  BC     0,0
*
*
*
*
*
*       DEBUG AID
*
          MVI    BUGM,C'M'
*
*
*
*
*
          SR     R2,R2
          IC     R2,0(R8,1)
          STH    R2,SCAN(R6)           STORE STATUS VALUE IN OUTPUT ARRAY
          LA     R6,2(R6)              UPDATE DISPLACEMENT TO SCAN ARRAY
          LA     R8,1(R8)              UPDATE DISPLACEMENT TO ANCILLARY
          C      R2,=F'0'              DATA,(WITHIN FILE)
          BNE    USESIT                CHECK IF CHANNEL IS ACTIVE
          ST     R3,CHNVL(R4)
          LA     R4,4(R4)
USESIT    BC     0,0
*
*       DEBUG AID
*
          MVI    BUGN,C'N'
*
          LA     R3,1(R3)
```

```
          BCT   R5,CHANSTAT          COMPLETE CHANSTAT LOOP
*
*       SKIP SCAN LINE NO. AS IT IS NOT 1 TO 1032 ON THE TAPE, BUT
*        SOME ARBITRARY NO. BETWEEN ANOTHER ARBITRARY NO.,N, AND
*        N+1031. THEREFORE LINE COUNT IS PERFORMED DYNAMICALLY IN
*        PROGRAM
*
CONTAN    BC    0,0                  'CONTINUE' FOR PASSING CONTROL
*
*       NEXT, VALUES OF LATITUDE,LONGITUDE,ETC. ARE RECORDED. NOTE,
*        ALL ARE RECORDED ON INPUT TAPE AS: SIGN(ONE BYTE),DEGREES,
*        (ONE BYTE),MINUTES(ONE BYTE), AND QUARTER SECONDS(ONE BYTE)
*        ALSO, LATER WE WILL CALCULATE ZENITH ANGLE FROM SUN ELEV.
*        ANGLE ACTUALLY FOUND ON TAPE
*
*       DEBUG AID
*
          MVI   BUGO,C'O'
*
          LA    R6,72                INITIALIZE DISPLACEMENT TO SCAN
*                                     ARRAY
          L     R8,POSINFIL          RE-INITIALIZE DISPLACEMENT TO R8
          LA    R8,70(R8)            GIVE ANCILLARY DISPLACEMENT,(POSIN-
*                                     FIL INCLUDES THE 2 BYTES FOR THE
*                                     FILE NO.)
          LA    R5,12                INITIALIZE INDEX TO ANGLES1 LOOP
*
ANGLES1   SR    R2,R2
          IC    R2,0(R8,1)           INSERT SIGN
          LTR   R2,R2
          BP    NEGVAL               IF VALUE IS 0001,NO. IS NEG.
          L     R2,=F'1'             IF VALUE IS 0000, NO. IS POS. SO
          B     ANGLESGN             MULTIPLYING FACTOR IS +1
NEGVAL    L     R2,=F'-1'            MULTIPLYING FACTOR FOR NEGATIVE
*                                     CASE IS -1.
ANGLESGN  ST    R2,SIGN              STORE MULTIPLYING FACTOR IN SIGN
          LA    R8,1(R8)             UPDATE DATA ACCESS VALUE
          LA    R7,3                 INITIALIZE INNER ANGLES2 LOOP
*                                     WHICH SECURES DEGREES,MINUTES AND
*                                     QUARTERSEC.
*
ANGLES2   SR    R2,R2
          SR    R3,R3
          IC    R3,0(R8,1)           INSERT VALUE
          M     R2,SIGN              MULTIPLY BY THE SIGN
```


REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
         STH   R3,SCAN(R6)          STORE IN SCAN ARRAY
         LA    R6,2(R6)            UPDATE DISPLACEMENT TO SCAN ARRAY
         LA    R8,1(R8)            UPDATE DISPLACEMENT TO INPUT DATA
         BCT   R7,ANGLES2          COMPLETE INNER ANGLES2 LOOP
*
         BCT   R5,ANGLES1          COMPLETE OUTER ANGLES1 LOOP
*
*        WE NOW CONVERT  ALL LONGITUDES TO WEST LONGITUDE,(THIS
*        IS THE FORM MOST FAMILIAR TO GSFC EXPERIMENTERS). VALUES
*        ON TAPE ARE 0 TO -180,(HEMISPHERE INCLUDING U.S.) AND
*        0 TO 180,(HEMISPHERE INCLUDING EUROPE).
*
         LA    R3,110
         LA    R5,4                THE FIRST THREE VALUES PROCESSED
*                                  ARE FROM THE IMAGE FOOTPRINT,THE
*                                  FOURTH IS THE SPACECRAFT VALUE,(THE
*                                  DISPLACEMENT,(IN R3) WILL BE MODIFIED
*                                  ACCORDINGLY
*
WLON     C     R5,=F'1'            IS THIS THE FOURTH CASE?
         BNE   WLONC               IF NOT, CONTINUE
         LA    R3,78               IF SO, INSERT DISPLACEMENT FOR
*                                  SPACECRAFT LONGITUDE.
WLONC    LH    R2,SCAN(R3)
         LTR   R2,R2
         BZ    ZLOND               IF ZERO DEG. CHECK THE MINUTES AT
*                                  ZLOND.
         BP    SUBW                IF POSITIVE, CONVERT,(360-ANGLE)
WDEGN    LPR   R2,R2               GET POSITIVE VALUE OF ANGLE
         STH   R2,SCAN(R3)
WDEGM    LH    R2,SCAN+2(R3)       CHANGE MINUTES
         LPR   R2,R2
         STH   R2,SCAN+2(R3)
WDEGS    LH    R2,SCAN+4(R3)       CHANGE SECONDS
         LPR   R2,R2
         STH   R2,SCAN+4(R3)
         LA    R3,12(R3)
         BCT   R5,WLON             COMPLETE WLON LOOP
         B     ZENCON              IF DONE,ZENITH CONVERSION IS NEXT
*
SUBW     L     R4,=F'240'          TO CALCULATE THE VALUE OF LONGIT.
         LH    R2,SCAN+4(R3)        SUBTRACT OSEC.,MIN.,AND DEG. RES-
         LPR   R2,R2               PECTFULLY FROM 240 OSEC.,59 MIN.,
         SR    R4,R2               AND 359. DEG. TO GET ACTUAL DIFFERENCE
         LH    R2,SCAN+2(R3)        INSERT ABSOLUTE VALUE OF MINUTES
```

```
         LPR   R2,R2
         C     R4,=F'240'
         BL    GLONS
         S     R4,=F'240'
         S     R2,=F'1'
GLONS    STH   R4,SCAN+4(R3)
         L     R4,=F'59'
         SR    R4,R2
         LH    R2,SCAN(R3)          INSERT ABSOLUTE VALUE OF
         LPR   R2,R2               DEGREES
         C     R4,=F'60'
         BL    GLONM
         S     R4,=F'60'
         S     R2,=F'1'
GLONM    STH   R4,SCAN+2(R3)        STORE CORRECTED MINUTES
         L     R4,=F'359'
         SR    R4,R2
         STH   R4,SCAN(R3)          STORE CORRECTED DEGREES
         LA    R3,12(R3)
         BCT   R5,WLON              COMPLETE WLON LOOP
         B     ZENCON               IF DONE, ZENITH CONVERSION IS NEXT
*
ZLOND    LH    R2,SCAN+2(R3)        TEST MINUTES
         LTR   R2,R2
         BZ    ZLONM
         BP    SUBW
         B     WDEGM
ZLONM    LH    R2,SCAN+4(R3)        TEST SECONDS
         LTR   R2,R2
         BZ    UPDATLON
         BP    SUBW
         B     WDEGS
UPDATLON LA    R3,12(R3)            UPDATE DISPL. TO LONGITUDE IN SCAN
         BCT   R5,WLON              COMPLETE WLON LOOP
*
*
ZENCON   BC    0,0
*
*       NOW CONVERT THE SUN ELEVATION ANGLE TO ZENITH ANGLE
*        VALUE IS 90 DEGREES MINUS SUN ELEVATION ANGLE
*
*
*       DEBUG AID
*
          MVI   BUGP,C'P'
```


REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
*
          LH     R2,SCAN+106          TO DERIVE DIFFERENCE BETWEEN SUN
          L      R3,=F'240'           ELEV. ANGLE AND 90. DEGREES,
          SR     R3,R2                SUBTRACT QSEC.,MIN.,DEG., FROM 240
*                                    QSEC.,59 MIN.,89 DEGREES YIELDING
          LH     R2,SCAN+104          ACTUAL DIFFERENCE.
          C      R3,=F'240'           CHECK THAT DIFFERENCE IS NOT GREATER
*                                    THAN SUBTRACTED VALUE.
          BL     GOODQSEC             IF NOT, CONTINUE
          S      R3,=F'240'           IF SO,SUBTRACT 240
          S      R2,=F'1'             SUBTRACT ONE,THEN,FROM MINUTES
GOODQSEC  STH    R3,SCAN+106          STORE CORRECTED VALUE OF QSEC FOR
*                                    ZENITH
          L      R3,=F'59'
          SR     R3,R2                SUBTRACT MINUTES FROM 59
          LH     R2,SCAN+102
          C      R3,=F'60'
          BL     GOODMIN
          S      R3,=F'60'            HERE,DIFFERENCE WAS GREATER THAN 60,
*                                    SO 60 IS SUBTRACTED AND DEGREES
          S      R2,=F'1'             VALUE IS DECREASED BY 1.
GOODMIN   STH    R3,SCAN+104          STORE CORRECTED VALUE OF MIN. FOR
*                                    ZENITH
          L      R3,=F'89'
          SR     R3,R2
          STH    R3,SCAN+102          STORE ZENITH DEGREES IN SCAN
*                                    ARRAY
*
*
*       DEBUG AID
*
          MVI    BUGO,C'0'
*
*
*       THE NEXT 21 PARAMETERS ARE MOVED DIRECTLY
*        (THEY ARE S-192 PARAMETER VALUES A1 THRU A21 AS
*         DESCRIBED IN PHO-TR 543)
*
*
          LA     R5,SCAN
          A      R5,=F'144'
          LR     R6,1
          A      R6,POSINFIL
          MVC    0(84,R5),118(R6)
          SR     R2,R2
          IC     R2,202(R6)           INSERT SCAN SHIFT DELAY FLAG
```

```
          LR     R3,R2                  TEST IF GIVEN BYTE WAS
          LA     R4,24                  NEGATIVE
          SLL    R3,0(R4)               IF NOT,LEAVE AS IS
          LTR    R3,R3                  IF SO, THE ONLY POSSIBLE
          BP     STORFLAG               VALUE IT COULD HAVE,(ACCORDING
          BZ     STORFLAG               TO THE HOUSTON DESCRIPTION), IS
          L      R2,=F'-1'              -1.INSERT THAT VALUE INTO SCAN.
STORFLAG  ST     R2,SCAN+228
          LH     R2,204(R6)
          ST     R2,SCAN+232            STORE AZIMUTH
          A      R5,=F'92'
          MVC    0(12,R5),248(R6)
*
*       THESE LAST THREE VALUES ARE RESPECTIVELY THE GROUND VELOCITY IN
*        METERS/SEC,THE ALTITUDE IN METERS AND THE ORBITAL INCLINATION
*        ANGLE IN QUARTER ARC SECONDS - THESE CONCLUDE THE ANCILLARY
*        BLOCK VALUES.
*
*       DEBUG AID
*
       MVI   BUGR,C'R'
*
          LA     R4,262                 UPDATE VALUE OF POSINFIL
          ST     R4,POSINFIL
*
*       THIS NEXT SECTION CONCERNS THE ACCESSING OF REGULAR DATA
*
LATERFIL  BC     0,0                    IT IS EXPECTED THAT THE ERRORS IN
*                                       MAGNITUDE ENCOUNTERED ON EARLIER
*                                       TAPES WILL HAVE BEEN CORRECTED ON
*                                       ALL TAPES RECEIVED AFTER 9/1/74 -
*                                       DATA IN UNITS OF
*                                       WATTS/CM**2/MICRON/STER
          L      R3,CHANNELO            DOUBLECHECK THAT CHANNEL NUMBER IS
          S      R3,=F'1'               ALL RIGHT
          SR     R2,R2
          M      R2,=F'4'
          L      R2,CHNVL(R3)
          LH     R4,SCAN
          CR     R2,R4
          BE     AOK
          STH    R2,SCAN
AOK       BC     0,0
          L      R8,POSINFIL            INITIALIZE DATA ACCESS REGISTER
          L      1,ADDAT                REINITIALIZE REG. 1 WITH
```


REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
*                                       ADDRESS OF DATA,(MAY DIFFER
*                                       IF DATA ACCESS IS NOT THE
*                                       FIRST FOR THAT PHYS. REC.)
*
*       DEBUG AID
*
          MVI    BUGS,C'S'
*
          LH     R5,HOUSEK+230          DET. TOTAL NO. PIXELS/CHANNEL
          LA     R7,248                 INITIALIZE OUTPUT INDEX FOR SCAN
          L      R6,SLDISPL              INITIALIZE SLOPE-INTERCEPT
*                                       DISPLACEMENT
*
PIXELS    SR     R2,R2
          IC     R2,0(R8,1)             INSERT PARAMETER VALUE
          FLOAT  R2,R2                  CHANGE TO FLOATING POINT
          ME     R2,SLIN+4(R6)          MULTIPLY BY SLOPE
          AE     R2,SLIN(R6)            ADD THE INTERCEPT
*                                       STORE IN SCAN OUTPUT ARRAY
*
          STE    R2,SCAN(R7)
*
*       DEBUG AID
*
          MVI    BUGT,C'T'
*
          LA     R8,1(R8)               UPDATE DATA ACCESS DISPLACEMENT
          LA     R7,4(R7)               UPDATE OUTPUT ARRAY DISPLACEMENT
          BCT    R5,PIXELS              COMPLETE PIXELS LOOP
*
          ST     R8,POSINFIL            STORE CURRENT VALUE OF R8,(DATA
*                                       ACCESS DISPLACEMENT) INTO POSINFIL
*
*       DEBUG AID
*
          MVI    BUGU,C'U'
*
          L      R2,NSCAN               DET. WHETHER LISTING OF SCAN DATA
          LTR    R2,R2                  IS DESIRED
          BZ     NOTWSHSC               IF NOT, SKIP CALL TO PRNTSC
*
*                                        NOTE THAT SINCE DATA IS LOADED
*                                       DIRECTLY INTO CALL PROGRAM AREA VIA
*                                       REG. R9, CALL ARGUMENTS ARE ALL
*                                       (R9) INSTEAD OF SCAN.ALSO,DATA
```

```
*                                       REFERENCE GIVEN 4 TIMES SINCE
*                                       VARIOUS TYPES OF DATA,(REAL,INTEGER,
*                                       ETC:) ARE FOUND IN SCAN.
          CALL   PRNTSC,((R9),(R9),(R9),(R9)),VL
*
NOTWSHSC  BC     0,0
*
*       RESTORE ALL BUT HOUSEKEEPING 'BUGS'
*
          MVI    BUGE,C'0'
          MVI    BUGF,C'0'
          MVI    BUGG,C'0'
          MVI    BUGH,C'0'
          MVI    BUGI,C'0'
          MVI    BUGJ,C'0'
          MVI    BUGK,C'0'
          MVI    BUGL,C'0'
          MVI    BUGM,C'0'
          MVI    BUGN,C'0'
          MVI    BUGO,C'0'
          MVI    BUGP,C'0'
          MVI    BUGQ,C'0'
          MVI    BUGR,C'0'
          MVI    BUGS,C'0'
          MVI    BUGT,C'0'
          MVI    BUGU,C'0'
*
          L      R2,NOCHANPR          DECREMENT THE NO. OF CHANNELS
          S      R2,=F'1'             LEFT IN THE PHYSICAL RECORD
          ST     R2,NOCHANPR
          LA     R4,0                 INSERT DATA TYPE INDICATOR,(REGULAR
*                                     DATA)
          L      R5,ADCALL            SECURE ADDRESS OF CALL ARGUMENTS
          L      R5,4(R5)
          ST     R4,0(R5)
*
          L      13,4(13)             RETURN PROCEDURE
          RETURN (14,12),T            RESTORE REGISTERS AND RETURN
*
*       ERROR EXIT!
*
ERRFILNO  LA     R4,1
          L      R5,ADCALL
          L      R5,4(R5)
          ST     R4,0(R5)
```

```
*
*       DEBUG AID
*
          MVI    BUGV,C'V'
*
          L      13,4(13)
          RETURN (14,12),T          RESTORE REGISTERS AND RETURN
*
*       END OF FILE EXIT!
*
ENDOFILE LA      R4,2
          L      R5,ADCALL
          L      R5,4(R5)
          ST     R4,0(R5)
*
*
*       DEBUG AID
*
          MVI    BUGW,C'W'
*
          L      13,4(13)           RETURN PROCEDURE
          RETURN (14,12),T
*
*-------------------------------------------------------------------
*
*       INITIALIZE VARIABLES AND STORAGE HERE
*
*         LTORG HERE PLACES ALL LITERALS REFERENCED IN PROGRAM HERE
*
          LTORG
*
*
*        THESE VARIABLES ARE INITIALIZED FOR DEBUG USE
*
BUG1      DC     C'0'
BUG2      DC     C'0'
BUG3      DC     C'0'
BUG4      DC     C'0'
BUG5      DC     C'0'
BUG6      DC     C'0'
BUG7      DC     C'0'
BUG8      DC     C'0'
BUG9      DC     C'0'
BUGA      DC     C'0'
BUGB      DC     C'0'
```


REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
BUGC     DC    C'0'
BUGD     DC    C'0'
BUGF     DC    C'0'
BUGF     DC    C'0'
BUGG     DC    C'0'
BUGH     DC    C'0'
BUGI     DC    C'0'
BUGJ     DC    C'0'
BUGK     DC    C'0'
BUGL     DC    C'0'
BUGM     DC    C'0'
BUGN     DC    C'0'
BUGO     DC    C'0'
BUGP     DC    C'0'
BUGQ     DC    C'0'
BUGR     DC    C'0'
BUGS     DC    C'0'
BUGT     DC    C'0'
BUGU     DC    C'0'
BUGV     DC    C'0'
BUGW     DC    C'0'
*
*
*
*      INPUT TAPE DESCRIPTION
*
TP192 DCB DSORG=PS,MACRF=(GL),DDNAME=TP192,EODAD=ENDOFILE,DEVD=TA
*
*
*      VARIABLES -
*
ADCALL   DC    F'0'                TEMPORARY STORAGE FOR ADDRESS OF
*                                  CALL ARGUMENTS
ADDAT    DC    F'0'                TEMPORARY STORAGE FOR ADDRESS OF
*                                  INPUT DATA
BLANK    DC    C'    '             TEMP. STORAGE FOR VALUES WHERE
*                                  BLANKS MUST PRECEDE
CHANNEL  DC    F'0'                WILL CONTAIN ACTUAL CHANNEL
*                                  NO. FOUND ON TAPE
CHANNELO DC    F'0'                WILL CONTAIN THE ORDER OF THE
CHNVL    DC    F'1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20'
         DC    F'21,22'            THIS IS THE ARRAY OF CHANNEL VALUE
*                                  INTO WHICH ACTUAL NAMES ON TAPE WILL
*                                  INSERTED.
CIND     DC    F'0'                TEMPORARY STORAGE AREA ALLOWING
```

```
*
FACTOR   DC    F'1.'      DOUBLE USE OF REGISTERS
*                         DATA MODIFYING FACTOR BY WHICH
*                         ALL DATA MUST BE MULTIPLIED -
*                         FOR ALL BUT THE 19TH CHANNEL, IT WILL
*                         BE 1000. FOR THAT ONE, IT IS 1.
FPT      DC    2F'0'      TEMPORARY STORAGE USED
*                         PRIMARILY FOR MACRO FLOAT
LINENO   DC    F'0'       STPRAGE FOR CURRENT LINE NO.,
*                          (IN ORDER) OF SCAN
NCOND    DC    F'0'       INDICATOR OF CONDITION -
*                         IF 0, GOOD DATA PRRSENT
*                        IF 1, ERROR WAS ENCOUNTERED, OR
*                         IF 2, AN END OF FILE WAS ENCOUNTERED
NHOUS    DC    F'0'        INDICATOR AS TO WHETHER USER
*                        DESIRES HOUSEKEEPING DATA LISTED
NOCHANPR DC    F'0'       CONTAINS THE NO. OF CHANNELS LEFT
*                         IN THE PHYSICAL RECORD
NOFIL    DC    F'0'       CONTAINS THE NO. OF FILES READ IN
*                         THE CURRENT DATA SET FOR THE CURRENT
*                         LINE.IT WILL BE LESS THAN OR EQUAL
*                         TO THE NO. OF FILES NEEDED TO
*                         COMPLETE ONE SCAN LINE FOR ALL
*                         CHANNELS,(AT PRESENT, 10).
NSCAN    DC    F'0'       INDICATOR AS TO WHETHER USER DESIRES
*                         SCAN DATA $ISTED
POSINFIL DC    F'0'       CONTAINS CURRENT POSITION IN
*                         CURRENT PHYSICAL RECORD
SAVE     DC    18F'0'      SAVE AREA FOR LINKAGE
SIGN     DC    F'0'        TEMPORARY STORAGE FOR SIGN OF
*                          ANGLES SUPPLIED IN THE ANCILLARY
*                          BLOCK
SLDISPL  DC    F'0'       DISPLACEMENT FOR ACCESSING SLOPE-
*                         INTERCEPT VALUES FOR CURRENT
*                         CHANNEL
TOTLINES DC    F'0'       TOTAL NO. OF LINES TO COMPLETE ONE
*                         SCAN
USEDIT   DC    F'0'       INDICATOR INITIALIZED TO ZERO,
*                         WHICH REVEALS WHETHER HOUSEKEEPING
*                         DATA HAS BEEN READ,(WHETHER FIRST
*                         TAPE ACCESS)
WORD     DC    F'0'       USED TO STORE A THREE BYTE VALUE
*                         TEMPORARILY - INSERTS LEADING ZERO
*                         THEREBY
*
*
```


REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

```
*       ARRAYS IN STORAGE
*
*
HOUSEK    DS     605F          HOUSEKEEPING ARRAY DATA
SLIN      DS     50F           SLOPE AND INTERCEPT ARRAY - NOTE
*                              REFERENCE TO IT CAN BE MADE WITH
*                              PROPER DISPLACEMENT FROM HOUSEK
SCAN      DSECT                 REFERENCE POINT FOR SCAN DATA,
*                               (STORAGE,VIA REG. 9,ACTUALLY IN
*                               CALLING PROGRAM)
*
*
*
          END
// EXEC SAVE
//LINK.OBJECT DD *
 NAME ZBMJMR92(R)
/*
/*
/*
```

```
//ZBFJMPL1 JOB (S10122604D,P,L00360,001003),B16,MSGLEVEL=1
//* THISDATE  L0011
// EXEC FORTRANH
//SOURCE.SYSIN DD *
C  THIS PROGRAM WAS WRITTEN BY F. MILASI OF CSC FOR TASK 092,
C  SKYLAB, UNDER CONTRACT NAS-5-11999 IN FEBRUARY, 1974.
C  IT UTILIZES THE WOLF PLOTTING AND CONTOURING PACKAGE, BY
C  READING IN THE INPUT DATA CARDS, AND CHECKING THEM AGAINST
C  THE OPTIONS THAT THE WOLF PLOT PACKAGE OFFERS.
C     ALSO DR. CURRAN (GSFC) HAS REQUESTED THAT THIS PROGRAM
C  SREACH THE INPUT TAPE (NEW S191) AND DO ONLY THOSE TIMES
C  THAT ARE REQUESTED THRU THE DATA CARDS. AFTER READING THE
C  DATA CARDS AND FINDING OUT WHICH DEVICE AND PLOTTER THAT
C  WAS REQUESTED, IT WILL THAN UTILIZES SUBROUTINE CLOUD,
C  PLOTT,CLDWRT
C*****************************************************************
C*****************************************************************
C
      COMMON/DAT91/DATUS(450),CLOUDS,REF(114),TBB(104),WRS191,NNN,
     1RRR(114),TTT(104)
      DIMENSION IVALE(5),IDEV(3),PLOTS(5),DEVIC(3),
     1RAD(114),WAVER(114),RADTEM(104),WAVET(104)
      EQUIVALENCE (DATUS(1),IDATA(1)),(DATUS(233),RAD(1)),
     1(DATUS(15),WAVER(1)),(DATUS(129),WAVET(1)),(DATUS(347),
     2RADTEM(1))
      REAL*8 PLOTER,PLOTS
      NAMELIST/INPUT/ PLOTER, IPRINT, MONTH, IDAY, IYEAR, IHOUR,
     1 MINUTE, ISEC, IENDHR, IENDMI, IENDSC
      DATA PLOTS/'CALCOMP ','SC4020FH','SC40208H','SC4020CF',
     1'PRINTER '/
      DATA IVALE /2000,0030,0020,0010,0001/
      DATA IPRINT/0/
      DATA MONTH/0/
      DATA ISEC/0/
      DATA IENDHR/0/
      DATA IENDMI/0/
      DATA IENDSC/0/
      DATA IDAY/0/
      DATA ITEAR/0/
      DATA IHOUR/0/
      INTEGER*2 IDATA(6)
      LOGICAL CHCLD/.TRUE./
      LOGICAL ENDCH/.FALSE./
      LOGICAL CHCLD/.FALSE./
      LOGICAL STPLD/.FALSE./
```


REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

```
      LOGICAL*4 CLOUDS
      LOGICAL*4 WRS191
      NNN=0
      READ(5, INPUT)
    5 CALL FREAD(DATUS,12,NOM,&160,&5)

      IF(ENDCH)GO TO 130
      IF(CHCLD)GO TO 120
      IF(MONTH.EQ.0)GO TO 90
      IF(MONTH.EQ.IDATA(1))GO TO 10
      WRITE(6,1000)MONTH,IDATA(1)
      STOP
   10 IF(IDAY.EQ.IDATA(2))GO TO 20
      WRITE(6,2000)IDAY,IDATA(2)
      STOP
   20 IF(IYEAR.EQ.IDATA(3))GO TO 30
      WRITE(6,3000)IYEAR,IDATA(3)
      STOP
   30 IF(IHOUR.EQ.IDATA(4))GO TO 40
      GO TO 5
   40 IF(MINUTE.EQ.IDATA(5))GO TO 50
      GO TO 5
   50 IF(ISEC.GE.IDATA(6))GO TO 55
      GO TO 5
   55 ENDCH=.TRUE.
      DO 70 I=1,5
      IF(PLOTER.EQ.PLOTS(I))GO TO 80
   70 CONTINUE
      GO TO 90
   80 NUM=IVALE(I)+IPRINT
      NUMM=4
      CALL NOSTAE
      CALL CPRIME(2)
      CALL PLOTST(NUM,NUMM)
      GO TO 120
   90 CHCLD=.TRUE.
      IF(STPLD)GO TO 120
      STPLD=.TRUE.
      NUM=0010
      NUMM=4
      CALL NOSTAE
      CALL CPRIME(2)
      CALL PLOTST(NUM,NUMM)
  120 CALL CLOUD
      GO TO 5
```

```
  130 IF(IDATA(4).LE.0.AND.IDATA(5).LE.0.AND.IDATA(6).LE.0)GO TO 160
      IF(IENDHR.GT.IDATA(4))GO TO 120
      IF(IENDHR.LT.IDATA(4))GO TO 160
      IF(IENDMI.GT.IDATA(5))GO TO 120
      IF(IENDMI.LT.IDATA(5))GO TO 160
      IF(IENDSC.GE.IDATA(6))GO TO 120
      WRITE(6,5000)MONTH,IDAY,IYEAR,IHOUR,MINUTE,ISEC,IENDHR,
     1IENDMI,IENDSC
  160 CALL SETBLK(999)
      CALL ENDPLT
      WRITE(6,4000)NNN
      WRITE(6, INPUT)
 1000 FORMAT(1H0,'PLEASE CHECK INPUT DATA ','MONTH',' DOES NOT AGREE WIT
     1H INPUT TAPE',10X,'MONTH=',1X,I4,5X,'IDATA(1)=',1X,I2)
 2000 FORMAT(1H0,'PLEASE CHECK INPUT DATA ','IDAY',' DOES NOT AGREE WITH
     1 INPUT TAPE ',10X,'IDAY=',1X,I4,5X,'IDATA(2)=',1X,I2)
 3000 FORMAT(1H0,'PLEASE CHECK INPUT DATA ','IYEAR',' DOES NOT AGREE WIT
     1H INPUT TAPE  ',10X,'IYEAR=',1X,I4,5X,'IDATA(3)=',1X,I2)
 4000 FORMAT(1H0,10X,'NUMBER OF FRAMES PLOTTED',2X,I4)
 5000 FORMAT(1H0,5X,'START TIME=',1X,I2,1X,I2,1X,I2,1X,I2,1X,I2,1X,I2,1X,
     1I2,5X,'END TIME=',1X,I2,1X,I2,1X,I2)
      STOP
      END
      SUBROUTINE CLOUD
      COMMON/DAT91/DATUS(450),CLOUDS,REF(114),TBB(104),WRS191,NNN,
     1RRR(114),TTT(104)
      DIMENSION RAD(114),WAVER(114),TEKVAL(114),RADTEM(104),
     1WAVET(104),DETA(55),DATAB(59),REFCOR(114)
      EQUIVALENCE (DATUS(1),IDATA(1)),(DATUS(233),RAD(1)),
     1(DATUS(15),WAVER(1)),(DATUS(129),WAVET(1)),(DATUS(347),RADTEM(1))
      EQUIVALENCE (TEKVAL(1),DETA(1))
      EQUIVALENCE (TEKVAL(56),DATAB(1))
      INTEGER*2 IDATA(6)
      DATA CONE/3.7413E+04/
      DATA CTWO/1.4388E+04/
      DATA PI/3.1416/
      DATA RADI/.017453/
      DATA DETA/
     X0.16721,0.16390,0.17500,0.17736,0.17464,
     X0.16922,0.16394,0.16663,0.18101,0.19219,
     X0.20083,0.20647,0.20332,0.20733,0.19471,
     X0.19414,0.18839,0.18340,0.18407,0.17824,
     X0.17271,0.16964,0.17125,0.17150,0.16996,
     X0.16657,0.16345,0.16020,0.15701,0.15439,
     X0.15111,0.14859,0.14560,0.14271,0.14018,
```


REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
     X0.13692,0.13439,0.13142,0.12898,0.12602,
     X0.12350,0.12110,0.11850,0.11590,0.11340,
     X0.11090,0.10855,0.10600,0.10360,0.10130,
     X0.09900,0.09680,0.09260,0.08910,0.08690/
      DATA DATAB/
     X0.08470,0.08200,0.07850,0.07480,0.07160,
     X0.06840,0.06530,0.06230,0.05930,0.05698,
     X0.05466,0.05250,0.05050,0.04850,0.04662,
     X0.04474,0.04298,0.04134,0.03970,0.03814,
     X0.03658,0.03538,0.03454,0.03370,0.03170,
     X0.02976,0.02880,0.02796,0.02712,0.02626,
     X0.02538,0.02450,0.02362,0.02274,0.02188,
     X0.02104,0.02020,0.01932,0.01844,0.01758,
     X0.01674,0.01590,0.01454,0.01324,0.01212,
     X0.01118,0.01030,0.00978,0.00926,0.00878,
     X0.00834,0.00790,0.00750,0.00710,0.00676,
     X0.00648,0.00620,0.00592,0.00564/
      DATA REFCOR/3.98,3.25,2.21,1.63,1.37,1.34,1.45,1.39,1.24,1.107,
     11.067,1.047,1.057,1.,1.057,1.067,1.028,1.,0.964,0.938,0.895,
     20.895,0.912,0.895,0.887,0.879,0.863,0.871,0.871,0.887,0.887,0.895,
     30.912,0.912,0.887,0.871,0.904,0.920,0.929,0.938,0.946,0.955,
     40.973,0.982,0.991,0.991,0.982,0.973,0.964,0.973,0.982,0.991,
     51.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.,1.018,1.,0.982,0.965,1.012,
     61.018,1.018,1.012,0.965,0.965,0.965,0.970,0.959,1.,1.,1.,1.,1.021,
     71.,1.008,1.025,1.042,1.042,1.047,1.042,1.021,1.,0.979,0.967,0.971,
     80.996,1.,1.,0.996,0.928,0.971,1.008,1.,1.,1.,1.,1.,0.898,0.883,
     90.932,0.890,0.872,0.812,0.857,1.056/
      NNN=NNN+1
      IF(DATUS(10).LT.0)GO TO 20
      DO 10 I=1,114
      ANG=DATUS(10)*RADI
      AC=COS(ANG)
      IF(AC.LE.0.)GO TO 15
      IF(TEKVAL(I).LE.0)GO TO 15
      REF(I)=(PI*RAD(I))/(AC*TEKVAL(I))
      REF(I)=REF(I)*REFCOR(I)
      RRR(I)=REF(I)
      IF(REF(I).LE.0.)REF(I)=0.
      IF(RRR(I).LE.0.)RRR(I)=0.
      IF(REF(I).GT.1.0)REF(I)=1.0
      GO TO 10
   15 REF(I)=0.
      RRR(I)=0.
   10 CONTINUE
   20 DO 30 K=1,104
```

```
      IF(RADTEM(K).LE.0.)GO TO 35
      TAB=1.+(CONE/(PI*((WAVET(K))**5)*RADTEM(K)))
      IF(TAB.LE.0.)GO TO 35
      TAA=WAVET(K)*ALOG(TAB)
      IF(TAA.LE.0.)GO TO 35
      TBB(K)=CTWO/TAA
      TTT(K)=TBB(K)
      IF(TBB(K).LT.190.)TBB(K)=190.
      IF(TBB(K).GT.310.)TBB(K)=310.
      GO TO 30
   35 TBB(K)=190.
      TTT(K)=0.
   30 CONTINUE
      CALL PLOTT
      RETURN
      END
      SUBROUTINE PLOTT
      COMMON/DAT91/DATUS(450),CLOUDS,REF(114),TBB(104),WRS191,NNN,
     1RRR(114),TTT(104)
      DIMENSION WAVER(114),WAVET(104)
      EQUIVALENCE (DATUS(1),IDATA(1)),(DATUS(15),WAVER(1)),
     1(DATUS(129),WAVET(1))
      LOGICAL*1 LABLL(14)/'CLOUD DATA XXX'/
      LOGICAL*1 LABEL(13)/'S191 DATA XXX'/
      LOGICAL*1 CH/' '/
      LOGICAL*4 CLOUDS
      LOGICAL*4 WRS191
      REAL*8 LAT,LONG,ZENITH
      INTEGER*2 IDATA(6)
      DATA RXHI,RXLO,RYHI,RYLO/2.6,0.4,1.0,0./
      DATA BXHI,BXLO,BYHI,BYLO/16.0,6.0,310.,190./
      CALL SETGRD(1.,1.,8.,4.0,4)
      CALL OGRID(RXLO,RXHI,11,'F4.1)',1,RYLO,RYHI,5,'F3.1)',1,0)
      IF(DATUS(10).LT.0)GO TO 10
      CALL PLOT(WAVER,REF,114,CH)
   10 CALL HORLIN('WAVELENGTH',10,5.0,0,-5,8)
      CALL VERLIN('REFLECTANCE',11,0.,2.5,4,-2)
      CALL SETGRD(1.,4.5,8.,7.5,4)
      CALL OGRID(BXLO,BXHI,10,'F4.1)',1,BYLO,BYHI,6,'F4.0)',1,0)
      CALL PLOT(WAVET,TBB,104,CH)
      CALL VERLIN('T B B',5,0.,6.5,4,-5)
      CALL HORLIN('MM DD YY',8,1.,8.0,0,1)
      CALL HORLIN('HH MM SS',8,2.5,8.0,5,1)
      CALL HORLIN('SUN ZENITH ANGLE',16,4.0,8.0,5,1)
      CALL HORLIN('LATITUDE',8,6.0,8.0,3,1)
```


REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
      CALL HORLIN('LONGITUDE',9,7.0,8.0,7,1)
      CALL INCORE(IDATA(1),IMONTH,18,1,2,0,0)
      CALL INCORE(IDATA(2),IDAY,18,1,2,0,0)
      CALL INCORE(IDATA(3),IYEAR,18,1,2,0,0)
      CALL INCORE(IDATA(4),IHOUR,18,1,2,0,0)
      CALL INCORE(IDATA(5),IMIN,18,1,2,0,0)
      CALL INCORE(IDATA(6),ISEC,18,1,2,0,0)
      CALL EDIT(DATUS(10),'F4.1/',ZENITH,NO,NO)
      CALL EDIT(DATUS(4),'F5.2/',LAT,NO,NO)
      CALL EDIT(DATUS(5),'F5.2/',LONG,NO,NO)
      CALL HORLIN(IMONTH,2,1.0,8.0,-5,-1)
      CALL HORLIN(IDAY,2,1.0,8.0,1,-1)
      CALL HORLIN(IYEAR,2,1.0,8.0,6,-1)
      CALL HORLIN(IHOUR,2,2.5,8.0,-1,-1)
      CALL HORLIN(IMIN,2,2.5,8.0,5,-1)
      CALL HORLIN(ISEC,2,2.5,8.0,11,-1)
      CALL HORLIN(ZENITH,4,4.0,8.0,5,-1)
      CALL HORLIN(LAT,5,6.0,8.0,5,-1)
      CALL HORLIN(LONG,5,7.0,8.0,8,-1)
      IF(CLOUDS)GO TO 20
      CALL EDIT(NNN,'I3/',LABEL(11),NO,NO)
      CALL HORLIN(LABEL,13,5.0,8.0,-4,4)
      IF(WRS191)GO TO 25
      GO TO 30
   20 CALL EDIT(NNN,'I3/',LABLL(12),NO,NO)
      CALL HORLIN(LABLL,14,5.0,8.0,-4,4)
   25 CALL CLDWRT
   30 CALL FRMADV
      RETURN
      END
      SUBROUTINE CLDWRT
      COMMON/DAT91/DATUS(450),CLOUDS,REF(114),TBB(104),WRS191,NNN,
     1RRR(114),TTT(104)
      DIMENSION  WAVER(114),WAVET(104)
      EQUIVALENCE (DATUS(1),IDATA(1)),(DATUS(15),WAVER(1)),
     1(DATUS(129),WAVET(1))
      LOGICAL*4 WRS191
      INTEGER*2 IDATA(6)
      IF(.NOT.CLOUDS)GO TO 10
      WRITE(6,1000)NNN
      GO TO 20
   10 WRITE(6,1500)NNN
   20 WRITE(6,2000)
      WRITE(6,3000)(IDATA(I),I=1,6),DATUS(10),DATUS(4),DATUS(5)
      WRITE(6,4000)
```

```
      IF(DATUS(10).LT.0)GO TO 10
      WRITE(6,5000) (WAVER(I) ,RRR(I) ,I=1,114)
   30 WRITE(6,6000)
      WRITE(6,7000)
      WRITE(6,8000)
      WRITE(6,9000) (WAVET(I) ,TTT(I) ,I=1,104)
 1000 FORMAT(1H1,T56,'CLOUD DATA',1X,I3)
 1500 FORMAT(1H1,T56,'S191 DATA',1X,I3)
 2000 FORMAT(1H0,4X,'MONTH',1X,'DAY',1X,'YEAR',3X,'HOUR',1X,'MINUTE',
     11X,'SECOND',13X,'SUN ZENITH ANGLE',16X,'LATITUDE',16X,'LONGITUDE')
 3000 FORMAT(1H ,6X,I2,3X,I2,2X,I2,5X,I2,4X,I2,6X,I2,21X,F4.1,22X,F5.2,
     221X,F5.2)
 4000 FORMAT(1H0,4X,'WAVELENGTH',6X,'REFLECTANCE',6X,'WAVELENGTH',
     16X,'REFLECTANCE',6X,'WAVELENGTH',6X,'REFLECTANCE',6X,
     2'WAVELENGTH',6X,'REFLECTANCE')
 5000 FORMAT(1H ,6X,F7.5,9X,F8.5,9X,F7.5,10X,F8.5,9X,F7.5,9X,F8.5,
     110X,F7.5,9X,F8.5)
 6000 FORMAT(1H ,)
 7000 FORMAT(1H ,20X,'TEMPERATURE',22X,'TEMPERATURE',21X,'TEMPERATURE',
     123X,'TEMPERATURE')
 8000 FORMAT(1H ,4X,'WAVELENGTH',6X,'BRIGHTNESS',7X,'WAVELENGTH',6X,
     1'BRIGHTNESS',6X,'WAVELENGTH',6X,'BRIGHTNESS',7X,'WAVELENGTH',
     27X,'BRIGHTNESS')
 9000 FORMAT(1H ,6X,F8.5,7X,F9.5,9X,F8.5,7X,F9.5,8X,F8.5,7X,F9.5,
     19X,F8.5,8X,F9.5)
      RETURN
      END
      BLOCK DATA
      COMMON/DAT91/DATUS(450),CLOUDS,REF(114),TBB(104),WRS191,NNN,
     1RRR(114),TTT(104)
      LOGICAL CLOUDS/.FALSE./
      LOGICAL WRS191/.TRUE./
      END
// EXEC LINKGO,REGION.GO=225K
//LINK.SYSLIB DD DSN=SYS2.WOLFPLOT,DISP=SHR
//GO.FT12F001 DD UNIT=(2400-9,,DEFER),DCB=(RECFM=VBS,
// LRECL=1804,BLKSIZE=30672,DEN=3),LABEL=(,BLP),
// VOL=SER=L0003
//GO.PLOTAPE DD DCB=(,DEN=1),LABEL=(,BLP,,OUT),UNIT=(7TRACK,,DEFER),
// VOL=SER=TEMSAV
//GO.WOLF4020 DD DCB=(,DEN=1),LABEL=(,BLP,,OUT),UNIT=(7TRACK,,DEFER),
// VOL=SER=TEMSAV
//GO.SYSUDUMP DD SYSOUT=A
//GO.DATA5 DD *
 &INPUT PLOTER='SC4020CF', IPRINT=1&END
```


REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

APPENDIX D - CLOUD S-191 PLOT PROGRAM LISTINGS, INPUT, AND JCL OUTPUT

```
IEF142I - STEP WAS EXECUTED - COND CODE 0000
IEF285I   SYS2.LOADLIB                                KEPT
IEF285I   VOL SER NOS= K3SYS4.
IEF285I   SYS2.WOLFPLOT                               KEPT
IEF285I   VOL SER NOS= M25YS2.
IEF285I   SYS2.DUMMY                                  KEPT
IEF285I   VOL SER NOS= K3SYS2.
IEF285I   SYS1.FORTLIB                                KEPT
IEF285I   VOL SER NOS= K3SYS6.
IEF285I   SYS2.FORTLIB                                KEPT
IEF285I   VOL SER NOS= K3SYS2.
IEF285I   SYS1.PL1LIB                                 KEPT
IEF285I   VOL SER NOS= K3SYS6.
IEF285I   SYS1.FORTSSP                                KEPT
IEF285I   VOL SER NOS= K3SYS2.
IEF285I   SYS2.COBLIB                                 KEPT
IEF285I   VOL SER NOS= K3SYS6.
IEF285I   SYS1.ALGLIB                                 KEPT
IEF285I   VOL SER NOS= K3SYS1.
IEF285I   SYS74212.T052325.RV00C.ZBFJMPL5.LOCMOD      PASSED
IEF285I   VOL SER NOS= K3SCR3.
IEF285I   SYS74212.T052325.SV000.ZBFJMPL5.R0C00SC7    SYSOUT
IEF285I   VOL SER NOS= K3SCR3.
IEF285I   SYS74212.T052325.RVCCC.ZBFJMPL5.R0000508    DELETED
IEF285I   VOL SER NOS= K3SCR2.
IEF285I   SYS74212.T052325.RV000.ZBFJMPL5.CBJMOD      DELETED
IEF285I   VOL SER NOS= K3SCR4.
IEF373I STEP /LINK     / START 74212.1911
IEF374I STEP /LINK     / STOP  74212.1912 CPU    0MIN 02.62SEC MAIN 130K LCS   0K
- STEP 02 -     RETURN CODE = 0000                                    STEP TIME =      .45 MINS=(CPU=      .04,IO=      .41)
                                    IO IN SECS, DISK=    24.65,DRUM=      .52,TAPE=      .00,CELL=      .00,OTHR=      .19
XXGO        EXEC PGM=*.LINK,SYSLMOD,COND=(4,LT),REGION=70K                  00023000
XXFT05F001 DD  DDNAME=DATA5                                                 00024000
XXFT06F001 DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=7265)               00025000
XXFT07F001 DD SYSOUT=B,DCB=(RECFM=FB,BLKSIZE=7280,LRECL=80)                 00026000
XXSYSPRINT DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=7265),              00027000
XX             SPACE=(CYL,(0,1)),UNIT=(DISK,3)                              00028000
//GO.FT12F001 DD UNIT=(2400-6,,DEFER),DCB=(RECFM=V3S,
// LRECL=1796,BLKSIZE=30536,DEN=3),LABEL=(,BLP),
// VOL=SER=Y-149
//GO.PLOTAPE DD DCB=(,DEN=1),LABEL=(,BLP,,OUT),UNIT=(7TRACK,,DEFER),
// VOL=SER=L004A
//GO.WOLF4020 DD DCB=(,DEN=1),LABEL=(,BLP,,OUT),UNIT=(7TRACK,,DEFER),
// VOL=SER=TEMSAV
//GO.SYSUDUMP DD SYSOUT=A
//GO.DATA5 DD *
//
IEF236I ALLOC. FOR ZBFJMPL5 GO
IEF237I 232   ALLOCATED TO PGM=*.DD
IEF237I 231   ALLOCATED TO FT05F001
IEF237I 332   ALLOCATED TO FT06F001
IEF237I 332   ALLOCATED TO FT07F001
IEF237I 332   ALLOCATED TO SYSPRINT
IEF237I 231   ALLOCATED TO SYSPRINT
IEF237I 232   ALLOCATED TO SYSPRINT
IEF237I 0C3   ALLOCATED TO FT12F001
IEF237I 0C0   ALLOCATED TO PLOTAPE
IEF237I 0D0   ALLOCATED TO WOLF4020
IEF237I 332   ALLOCATED TO SYSUDUMP
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
IEF142I - STEP WAS EXECUTED - COND CODE 0004
IEF285I   SYS2.TRANSLAT                                  KEPT
IEF285I   VOL SER NOS= K3SYS5.
IEF285I   SYS74212.T052325.RV000.ZBFJMPL5.OBJMOD         PASSED
IEF285I   VOL SER NOS= K3SCR4.
IEF285I   SYS74212.T052325.RV000.ZBFJMPL5.R0C00501       SYSOUT
IEF285I   VOL SER NOS= K3SCR4.
IEF285I   SYS74212.T052325.RV000.ZBFJMPL5.R0C00502       DELETED
IEF285I   VOL SER NOS= K3SCR4.
IEF285I   SYS74212.T052325.RV000.ZBFJMPL5.R0000503       DELETED
IEF285I   VOL SER NOS= K3SCR5.
IEF285I   SYS74212.T052325.RV00C.ZBFJMPL5.R0C00504       DELETED
IEF285I   VOL SER NOS= K3SCR3.
IEF285I   SYS74212.T052325.RV00C.ZBFJMPL5.R0C00505       DELETED
IEF285I   VOL SER NOS= K3SCR5.
IEF285I   SYS74212.T052325.RV00C.ZBFJMPL5.S0C00506       SYSIN
IEF285I   VOL SER NOS= K3SCR3.
IEF285I   SYS74212.T052325.RV000.ZBFJMPL5.S0C00SC6       DELETED
IEF285I   VOL SER NOS= K3SCR3.
IEF373I STEP /SOURCE  / START 74212.1902
IEF374I STEP /SOURCE  / STOP  74212.1911 CPU    CMIN 10.02SEC MAIN 248K LCS   0K
- STEP 01 -     RETURN CODE = 0004                                  STEP TIME =     .84 MINS=(CPU=     .16,IO=     .68)
                                        10 IN SECS, DISK=    40.05,DRUM=     .92,TAPE=     .00,CELL=     .00,OTHR=     .15
//  EXEC LINKGO,REGION.GO=275K
XXDEFAULT PROC NBLK=40                    PROC=LINKGO  DATE=72277          00001000
XXLINK   EXEC  PGM=IEWL,PARM=(MAP,LIST),COND=(4,LT),REGION=150K            00002000
XXLOADLIB  DD  DSNAME=SYS2.LOADLIB,DISP=SHR                                00003000
XXNEWLIN   DD  DUMMY                                                       00004000
//LINK.SYSLIB DD DSN=SYS2.WOLFPLOT,DISP=SHR
XX*SYSLIB    DD  DSNAME=SYS2.DUMMY,DISP=SHR                                00005000
XX           DD  DSNAME=SYS2.DUMMY,DISP=SHR                                00006000
XX           DD  DSNAME=SYS1.FORTLIB,DISP=SHR                              00007000
XX           DD  DSNAME=SYS2.FORTLIB,DISP=SHR                              00008000
XX           DD  DSNAME=SYS1.PLILIB,DISP=SHR                               00009000
XX           DD  DSNAME=SYS1.FORTSSP,DISP=SHR                              00010000
XX           DD  DSNAME=SYS2.COBLIB,DISP=SHR                               00011000
XX           DD  DSNAME=SYS1.ALGLIB,DISP=SHR                               00012000
XXSYSLMOD  DD  DSNAME=&&LODMOD(CSEC),DISP=(NEW,PASS),                      *00013000
XX             SPACE=(6144,(&NBLK,20,1)),UNIT=DISK                         00014000
IEF653I SUBSTITUTION JCL - SPACE=(6144,(40,20,1)),UNIT=DISK
XX*SYSPRINT DD SYSOUT=A,DCB=(RECFM=FBM,LRECL=121,BLKSIZE=3509)             00016000
XX*SYSUT1    DD  UNIT=(DISK,SEP=(SYSLMOD,SYSPRINT)),SPACE=(TRK,(20,5))     00017000
XX*TAPELIB   DD  DUMMY,DISP=(OLD,KEEP),UNIT=(9TRACK,,DEFER),              *00018000
XX               LABEL=(,BLP),                                             *00019000
XX               DCB=(,RECFM=FB,LRECL=80,BLKSIZE=3200)                     00020000
XX*SYSLIN DD DSNAME=&&OBJMOD,DISP=(OLD,DELETE),DCB=RECFM=FB                00021000
XX           DD  DDNAME=OBJECT                                             00022000
IEF236I ALLOC. FOR ZBFJMPL5 LINK
IEF237I 337   ALLOCATED TO LOADLIB
IEF237I 326   ALLOCATED TO SYSLIB
IEF237I 335   ALLOCATED TO
IEF237I 334   ALLOCATED TO
IEF237I 335   ALLOCATED TO
IEF237I 334   ALLOCATED TO
IEF237I 335   ALLOCATED TO
IEF237I 334   ALLOCATED TO
IEF237I 336   ALLOCATED TO
IEF237I 232   ALLOCATED TO SYSLMOD
IEF237I 232   ALLOCATED TO SYSPRINT
IEF237I 231   ALLOCATED TO SYSUT1
IEF237I 331   ALLOCATED TO SYSLIN
```

```
F128-LEVEL LINKAGE EDITOR OPTIONS SPECIFIED MAP,LIST
        DEFAULT OPTION(S) USED - SIZE=(131072,12288)
```

MODULE MAP

CONTROL SECTION NAME		ORIGIN	LENGTH
MAIN		00	738
CLOUD		738	46A
PLOTT		BA8	BA4
CLOWRT		1450	496
DATA1		1908	DD4
IHCSLOG	*	26E0	196
PLOTST	*	3898	9C4
VECTOR	*	3660	23A
IHCNAMEL	*	41F8	ACP
FREAD	*	4CC8	1A67
FUNITABL	*	6730	CP
FPL1	*	67F8	4
IHCECOMH	*	6800	F61
IHCCOMH2	*	7768	65D
NOSTAE	*	7DC8	9C
CALCMP	*	7E68	404C
IHCSSCN	*	BEB8	109
EDIT	*	C098	6DA
DIAGLN	*	C770	1100
IHCERRM	*	D890	5DA
INCORE	*	DE58	27C
GRID	*	E0C8	1082
PLOT	*	F150	98E
AFFINE	*	F940	1204
UCS	*	10D48	456
IHCFCVTH	*	111A0	1190

ENTRY NAME	LOCATION	NAME	LOCATION	NAME	LOCATION	NAME	LOCATION
ALOG10	26F0	ALOG	26F8				
CALSIZ	13F0	CPRIME	3430	DEVSET	348C	SETGRD	3508
SAMSIZ	15B4	SETLET	15E0				
ENDPLT	4128	FRMADV	416C	PAWS	4180		
FRCNL#	41F8	FWRNL#	47FC				
FREADB	4EFC	FWRITE	4F80	REWIND	5106	LEAVE	527A
UNLOAD	531E	FOSN	5424	MOUNT	561C	MEMBER	570A
IRCOM#	6900	FDIOCS#	68BC	INTSWTCH	7746		
SEQ0ASD	7AE0						
CTRANS	BD94	SETBLK	BE00	BLKADR	BE54		
COS	BEB8	SIN	BED0				
HORLIN	051C	VERLIN	05CC	RANCHR	067C	TYPOFF	0730
TICMRK	0780	LSEQEV	0820				
ERRMCN	0880	IHCERRE	0898				
CGRID	F028	PLABEL	F108				
COORD	FA40	NEGLOG	FAD8				
DATADJ	10A58	GENPLT	10AC8	TCOORD	10B24	SCALE	10B88
POLAR	10C64	PSTGRD	10C84				

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

NAME	ORIGIN	LENGTH
IHCEFNTH*	12340	542
IHCEFIOS*	12888	F28
IHCFIOS2*	13780	52E
IHCFIXPI*	13CE0	14F
GERBER *	13F30	F4C
PRINTR *	14D80	4CC4
SC4020 *	19448	184C
SD4060 *	18598	1100
CHARPL *	1C698	D20
CLPLTS *	1D388	9D8
IHCFRXPI*	1DD90	141
GRBTAP *	1E0D8	17C
IHCETRCH*	1E058	29E
IHCUATBL*	1E2E8	638
IHCUOPT *	1E920	328
MINT *	1EC48	12F
SCHAR *	1ED78	11A
IHCSSQRT*	1EE98	145
SCTAP *	1EFE0	17C
SDTAP *	1F160	17C
CPLOT$	1F2E0	1E8
CON$	1F4C8	118
FERMSG *	1F5E0	68

ENTRY ADDRESS 00
TOTAL LENGTH 1F648

NAME	LOCATION	NAME	LOCATION	NAME	LOCATION	NAME	LOCATION
ADCON#	111A0	FCVACUTP	1124A	FCVL OUTP	1120A	FCVZOUTP	1142A
FCVICUTP	11708	FCVEOUTP	11CDA	FCVCOUTP	11EF4	INT6SWCH	12108
ARITH#	12340	ADJSWTCH	1260C				
FIOCS#	12889	FIOCSBEP	1289E				
FIXPI#	13CE0						
GPEN	14C0C	GTRANS	14C5C	PSTOPS	14CC4	PTERMS	14D20
CONDNS	19848	EXPAND	19890	PRUNIT	198D8	PAGECT	1992C
PRCHAR	1998C	SIMSC	199F0				
FRAMES	18534						
SDCOPY	1C514	SDFRMS	1C564	SDERRS	1C5C0	SDMODE	1C610
ROCHAR	1D188	CSET	1D21C	INTENS	1D280	RUTLET	1D300
SETSIZ	1D35C						
OFFSET	1D474	ZIP	1D44A	FACTOR	1D4E4	WHERE	1D502
CALPLT	1D524						
FRXPI#	1DD90						
IHCTRCH	1E058	CRRTRA	1E060				
SQRT	1EE98						

****GSFC DOES NOT EXIST BUT HAS BEEN ADDED TO DATA SET

S191 DATA 1

MONTH	DAY	YEAR	HOUR	MINUTE	SECOND	SUN ZENITH ANGLE	LATITUDE	LONGITUDE
9	3	73	18	11	20	0.0	16.08	83.50

WAVELENGTH	REFLECTANCE	WAVELENGTH	REFLECTANCE	WAVELENGTH	REFLECTANCE	WAVELENGTH	REFLECTANCE
0.40000	0.07127	0.40500	0.08903	0.41000	0.11938	0.41500	0.16048
0.42000	0.17018	0.42500	0.16849	0.43000	0.16461	0.43500	0.17034
0.44000	0.17520	0.44500	0.18226	0.45000	0.19100	0.46000	0.19545
0.47000	0.19894	0.48000	0.20206	0.49000	0.20193	0.50000	0.19232
0.51000	0.18852	0.52000	0.18669	0.53000	0.18151	0.54000	0.18040
0.55000	0.18181	0.56000	0.17362	0.57000	0.16107	0.58000	0.14719
0.59000	0.13715	0.60000	0.13664	0.61000	0.13968	0.62000	0.13698
0.63000	0.13576	0.64000	0.13450	0.65000	0.14075	0.66000	0.14314
0.67000	0.13723	0.68000	0.13032	0.69000	0.12617	0.70000	0.11633
0.71000	0.11507	0.72000	0.12765	0.73000	0.13372	0.74000	0.13711
0.75000	0.13546	0.76000	0.13646	0.77000	0.14197	0.78000	0.14488
0.79000	0.14212	0.80000	0.13413	0.81000	0.12532	0.82000	0.12122
0.83000	0.12454	0.84000	0.12149	0.85000	0.13423	0.86000	0.13598
0.88000	0.12241	0.90000	0.09855	0.92000	0.05692	0.94000	0.03156
0.96000	0.05974	1.00000	0.07671	1.02000	0.08134	1.04000	0.08275
1.06000	0.07419	1.08000	0.06840	1.10000	0.06887	1.12000	0.01599
1.14000	0.02012	1.16000	0.05085	1.18000	0.07341	1.20000	0.08356
1.22000	0.07973	1.24000	0.05831	1.26000	0.10314	1.28000	0.08587
1.30000	0.06252	1.32000	0.03789	1.34000	0.01288	1.36000	0.0
1.38000	0.0	1.40000	0.0	1.44000	0.01090	1.48000	0.04539
1.50000	0.07527	1.52000	0.09663	1.54000	0.09962	1.56000	0.10408
1.58000	0.10893	1.60000	0.10643	1.62000	0.10773	1.64000	0.10638
1.66000	0.11199	1.68000	0.10900	1.70000	0.09798	1.72000	0.09757
1.74000	0.07667	1.76000	0.05718	1.78000	0.02627	1.80000	0.00790
1.94000	0.0	1.88000	0.0	1.92000	0.01037	1.96000	0.01967
2.00000	0.03660	2.04000	0.06103	2.08000	0.09160	2.12000	0.10019
2.16000	0.09664	2.20000	0.10339	2.24000	0.09634	2.28000	0.04425
2.32000	0.04183	2.36000	0.01939	2.40000	0.0	2.44000	0.0
2.48000	0.0						

WAVELENGTH	TEMPERATURE BRIGHTNESS	WAVELENGTH	TEMPERATURE BRIGHTNESS	WAVELENGTH	TEMPERATURE BRIGHTNESS	WAVELENGTH	TEMPERATURE BRIGHTNESS
6.00000	0.0	6.10000	0.0	6.20000	0.0	6.30000	0.0
6.40000	0.0	6.50000	0.0	6.60000	0.0	6.70000	0.0
6.80000	0.0	6.90000	0.0	7.00000	195.26064	7.10000	0.0
7.20000	227.69336	7.30000	233.49510	7.40000	242.85533	7.50000	246.60164
7.60000	249.62954	7.70000	248.52879	7.80000	255.24376	7.90000	258.93511
8.00000	266.71826	8.10000	274.34546	8.20000	284.16553	8.30000	286.22705
8.40000	290.04932	8.50000	289.84668	8.60000	290.44092	8.70000	291.69531
8.80000	291.24365	8.90000	292.69800	9.00000	292.88794	9.10000	292.80566
9.20000	292.06519	9.30000	288.80225	9.40000	280.39819	9.50000	0.0
9.60000	269.10400	9.70000	0.0	9.80000	274.13477	9.90000	282.09033
10.00030	288.89526	10.10000	292.84326	10.20000	294.20776	10.30000	294.93359
10.40000	294.18457	10.50000	294.41992	10.60000	294.96533	10.70000	294.97583
10.80000	295.00830	10.90000	294.55029	11.00000	294.39673	11.10000	294.48389
11.20000	294.13306	11.30000	293.96694	11.40000	294.00439	11.50000	293.60156
11.60000	293.03760	11.70000	292.55249	11.80000	292.23486	11.90000	292.43555
12.00000	292.56738	12.10000	293.15430	12.20000	292.42969	12.30000	291.23169
12.40000	289.71338	12.50000	288.62354	12.60000	287.20410	12.70000	286.74561
12.80000	287.80884	12.90000	287.67261	13.00000	287.01367	13.10000	283.84912
13.20000	279.83521	13.30000	274.98120	13.40000	269.41797	13.50000	264.64819
13.60000	261.23999	13.70000	261.51685	13.80000	258.18555	13.90000	254.51869
13.95000	253.46869	14.00000	252.67202	14.05000	252.26619	14.10000	251.16473
14.15000	249.37749	14.20000	247.19558	14.25000	244.42529	14.30000	240.63127
14.35000	236.64775	14.40000	232.85443	14.45000	229.70036	14.50000	226.61943
14.55000	224.07295	14.60000	221.19402	14.65000	219.38971	14.70000	217.03844
14.75000	215.10438	14.80000	214.40636	14.90000	213.50058	15.00000	212.57844
15.10000	213.78865	15.20000	207.82047	15.30000	201.56413	15.40000	192.46129

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

APPENDIX E - INTERPOLATION PROGRAMS CNTH91 AND CNTH92 PROGRAM LISTINGS, INPUT, AND JCL OUTPUT

```
//ZBMJM001 JOB (S10122604D,T,L00360,002002),B16
//* THISDATE
// EXEC FORTRANH,PARM='OPT=2,XREF,ID,MAP,EBCDIC',REGION=300K
//* NOTE - DEFAULT REGION ON 75 AND 91 AFTER 6/12/74 IS ONLY 250K FOR
//* FORTRANH AND 150K FOR FORTRANG. MOST OF SKYLAB PROGRAMS USING
//* JCLFORT NEED 300K.
//SOURCE.SYSIN DD *
C       THIS PROGRAM WAS WRITTEN BY M. MACK OF C.S.C.
C       FOR TASK 092,SKYLAB,UNDER CONTRACT NAS-5-11999
C       IN JANUARY,1974. IT USES A GAUSSIAN FILTER TO
C       'AVERAGE' SOLAR IRRADIANCE VALUES, SUPPLIED
C       ORIGINALLY BY DR. THEKAKARA THRU DR. CURRAN
C       (GSFC), AND CONVERTED TO COMPATIBLE UNITS
C       UNDER THE SAME TASK 092 BY PROGRAM 'CON'. THESE
C       'AVERAGES' ARE DERIVED CORRESPONDING TO EACH OF
C       THE KNOWN SKYLAB S-191 WAVELENGTHS FROM .4 MICRONS
C       TO .74 MICRONS. FOR WAVELENGTHS BETWEEN .75 MICRONS
C       AND 2.5 MICRONS,(SINCE THE FILTER'SHAPE' DOES NOT
C       AFFECT THE IRRADIANCE VALUE ACCORDING TO DR. CURRAN),
C       A STRAIGHT LINE (LINEAR) INTERPOLATION IS PERFORMED.
C       LATER ANALYSIS USE OF THESE VALUES WILL BE
```

```
C       CONFINED TO THE .4 TO 2.5 REGION SO OTHER
C       KNOWN SKYLAB S-191 WAVELENGTHS WILL NOT
C       BE CONSIDERED.
C       CHANGED FROM NOPTS=113 TO NOPTS=114 IN JAN.'75 TO
C       ACCOMODATE NEW DATA.(NEITHER SET A PERFECT SUBSET
C       OF THE OTHER-I.E. ONLY OVERLAPPING).
C
      COMMON /DATATH/ WVTH(3297),AIRTH(3297)
      DIMENSION WAVSKY(114),AIRSKY(114)
C
C       LOGICAL VARIABLE USED IN CODING
C
      LOGICAL DO1
      DATA NOTOAV,NOTOIN,NOPTS,NPTS/40,41,114,3297/
C
C     NOTOAV IS THE NUMBER OF WAVELENGTHS WHICH MUST
C            BE 'AVERAGED' USING THE GAUSSIAN FILTER.
C     NOTOIN IS THE BEGINNING ELEMENT NUMBER IN WAVSKY
C            AT WHICH LINEAR INTERPOLATION WILL BE USED.
C            (IT WILL BE NOTOAV+1)
C     NOPTS  IS THE NUMBER OF WAVELENGTHS ALTOGETHER
C            IN WAVSKY
C     NPTS   IS THE NUMBER OF WAVELENGTHS COVERED IN
C            DR. THEKAKARA'S DATA
C
      DATA SIGMAF / 0.00625/
C
C     SIGMAF IS THE FACTOR BY WHICH EACH WAVELENGTH
C            IS MULTIPLIED TO DETERMINE THE 'SIGMA'
C            VALUE FOR THE FILTER USED FOR THAT
C            WAVELENGTH
C       IT IS ACTUALLY EQUAL TO THE FILTER HALFWIDTH AT
C       HALF MAXIMUM,(I.E. TRANSMISSION IS ROUGHLY
C       EQUAL TO A*EXP(-X/SIGMA) WHERE A IS AN
C       ARBITRARY CONSTANT - SIGMAF WOULD BE THE
C       VALUE WHICH WOULD YIELD HALF THE
C       MAXIMUM TRANSMISSION VALUE)
C      (WHOLE WIDTH GIVEN BY DR. CURRAN WAS
C      EQUAL TO 0.0125 MICRONS - WE USE HALF!)
C
C
C       FORMAT STATEMENTS
C
 1000 FORMAT(5F10.0)
 1500 FORMAT('THE FOLLOWING ARE THE ORDERED WAVELENGTHS',
```


REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
     1' FOR SKYLAB'/,'DATA - USED IN CONJUNCTION WITH',
     2' CALCULATED-ASSOCIATED'/,'THEKAKARA DATA - S-191')
1750 FORMAT(1H1, 'INPUT VALUES'/)
2000 FORMAT(5X,'X',F7.3,',',F7.3,',',F7.3,',',F7.3,
     1 ',',F7.3,',')
2500 FORMAT('END OF WAVELENGTH DATA - BEGINNING OF',
     1' CALCULATED-ASSOCIATED THEKAKARA'/,
     2'DATA - S191 COMPARISON')
3000 FORMAT(5X,'X',F7.5,',',F7.5,',',F7.5,',',F7.5,
     1 ',',F7.5,',')
3500 FORMAT(1H1,'DATA CALCULATED AND OUTPUT TO UNIT',
     1 ' 7'/,1X,3(7X,'WAVELENGTH',4X,'IRRADIANCE VALUE',
     2 5X)//)
4000 FORMAT(1H ,I4,'.',2X,F10.6,8X,F9.7,8X,I4,'.',2X,
     1F10.6,8X,F9.7,8X,I4,'.',2X,F10.6,8X,F9.7)
4500 FORMAT(1HO,'END OF PROGRAM - VALUES CALCULATED AND'
     1' PRINTED')
6000 FORMAT(1H ,I5,4E16.8,3I5/,3X,5E16.8)
7000 FORMAT(1H ,I5,F10.5,2I6,4E16.8)
C
C       AS THE FOLLOWING VALUE IS USED FREQUENTLY, BUT IS
C       A CONSTANT, AND AS WE WISH THE LIMIT OF MACHINE
C       ACCURACY,(AS WELL AS CONSISTENCY IN MACHINE
C       CALCULATION), WE CALCULATE IT HERE.
C
      ALN2=ALOG(2.)
C
C       WE NOW READ IN THE ARRAY OF DESIRED SKYLAB S-191
C       COMPARABLE WAVELENGTHS
C
      READ(5,1000) WAVSKY
C
C       THIS FIRST LOOP WILL CALCULATE THE 'AVERAGE' DESIRED FOR
C       WAVELENGTHS BETWEEN .4 AND .74 MICRONS - OUTER
C       LOOP WILL DETERMINE LIMITS OF INNER LOOP
C
      DO 400 LB=1,NOTOAV
      WAV=WAVSKY(LB)
      SIG=SIGMAF*WAV
C
C       HERE, THE 'SPREAD' AMONG THEKAKARA'S DATA FOR
C       NUMERICAL INTERPOLATION IS DETERMINED,(I.E.,HERE,
C       THE LIMITS OF THE INNER LOOP)
C
      W1=WAV-SIG
```

```
      W2=WAV+SIG
      DO1=.TRUE.
C
      DO 150 LZ=1,NPTS
      IF(.NOT. DO1) GO TO 100
      IF(W1.LT.WVTH(1)) GO TO 95
      IF(W1.GT. WVTH(LZ)) GO TO 150
      I1=LZ-1
      DO1=.FALSE.
      IF(I1.LT.1) I1=1
      GO TO 100
   95 CONTINUE
      I1=1
      DO1=.FALSE.
  100 CONTINUE
      IF(W2.GT.WVTH(LZ)) GO TO 150
      I2=LZ+1
      GO TO 200
  150 CONTINUE
C
  200 CONTINUE
      IF(I2.GT.NPTS) I2=NPTS
C
C      NOW THE 'AVERAGE' MAY BE CALCULATED(NUMERICAL
C      INTERPOLATION PERFORMED)
C
      SIGPRD =ALN2/(SIG*SIG)
      SUM1=0.
      SUM2=0.
      DO 300 LTR=I1,I2
C
C      FIRST CALSULATE WAVELENGTH DIFFERENCE
C
      WD=WVTH(LTR)-WAV
C
C
C      NOW CALCULATE -((NU-NUO)/SIGMA)**2 * LN2
C
      WD2P=-WD*WD*SIGPRD
C
C      TAKE EXPONENTIAL OF ABOVE
C
      WEXP=EXP(WD2P)
      SUM1=SUM1+WEXP
C
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
C       TAKE PRODUCT OF EXPONENTIAL VALUE AND IRRADIANCE
        ARWEXP=AIRTH(LTR)*WEXP
        SUM2=SUM2+ARWEXP
        WRITE(6,6000) LB,WAV,W1,W2,SIG,LTR,I1,I2,
       1WD,WD2P,SUM1,ARWEXP,SUM2
C
  300 CONTINUE
      AIRSKY(LB)=SUM2/SUM1
  400 CONTINUE
C
C       ON THE DATA BETWEEN .75 AND 2.5 MICRONS, WE
C       NOW PERFORM STRAIGHT LINE INTERPOLATION
C
      DO 800 LD=NOTOIN,NOPTS
      WAV=WAVSKY(LD)
      IF(WAV.LT. WVTH(1)) GO TO 550
C
C       FIND 'SURROUNDING' POINTS
C
      DO 500 LZ=1,NPTS
      J1=LZ-1
      J2=LZ
      IF(WAV.GT. WVTH(LZ)) GO TO 500
      IF(J1.LT. 1) J1=1
      IF(WAV.EQ. WVTH(J1)) GO TO 700
      IF(WAV.EQ. WVTH(J2)) GO TO 650
      GO TO 600
  500 CONTINUE
C
C
C       VALUE IS BEYOND RANGE OF DATA - SET TO LAST TWO POINTS
C         FOR EXTRAPOLATION
C
      J1=NPTS-1
      J2=NPTS
      GO TO 600
C
  550 CONTINUE
C
C       DATA VALUE LIES BEFORE BEGINNING OF INPUT DATA
C         POINTS - SET TO FIRST TWO POINTS FOR EXTRAPOLATION
C
      J1=1
      J2=2
  600 CONTINUE
```

```
C
C       WE MUST INSURE THAT WVTH VALUES ARE NOT THE SAME,(SOME ARE)
C
  625 CONTINUE
      IF(WVTH(J2).NE. WVTH(J1)) GO TO 635
      J2=J2+1
      GO TO 625
  635 CONTINUE
C
      IF(J2.GT.NPTS) J2=NPTS
C
      WVDF1=WVTH(J2)-WVTH(J1)
      WVDF2=WAV-WVTH(J1)
      AIRDF=AIRTH(J2)-AIRTH(J1)
      AIRSKY(LD)=AIRTH(J1)+(AIRDF/WVDF1)*WVDF2
      WRITE(6,7000) LD,WAV,J1,J2,WVDF1,WVDF2,AIRDF,
     1AIRSKY(LD)
C
      GO TO 800
  650 CONTINUE
      J1=J2
  700 CONTINUE
      AIRSKY(LD)=AIRTH(J1)
  800 CONTINUE
C
C       INPUT VALUES ARE LISTED FOR CHECKING PURPOSES
C
      WRITE(9,1750)
      WRITE(9,4000) (LX,WVTH(LX),AIRTH(LX),LX=1,NPTS)
C
C       WE NEXT PRINT OUT RESULTS BOTH ON
C       CARDS ,(FOR FUTURE USE), AND ON UNIT '9',(SYSOUT A),
C       FOR SCIENTISTS TO STUDY.
C
C         THIS LOOP WAS INSERTED TO CREATE SEVERAL DECKS
C           AT ONCE WITHOUT WAITING FOR IBM 20 TURN-AROUND.
      DO 900 LDECK=1,4
      WRITE(7,1500)
      WRITE(7,2000)WAVSKY
      WRITE(7,2500)
      WRITE(7,3000) AIRSKY
C
C         END OF DECK LOOP
  900 CONTINUE
      WRITE(9,3500)
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
      WRITE(9,4000) (LD,WAVSKY(LD),AIRSKY(LD),LD=1,NOPTS)
      WRITE(6,4500)
      RETURN
      END
// EXEC LINKGO,REGION.GO=200K
//LINK.OBJECT DD *
 INCLUDE LOADLIB(ZBMJMCRB)
//*     UNIT 6 IS THE DEBUG AND GENERAL MESSAGE UNIT ON PRINTER
//*      UNIT 7 IS THE CARD OUTPUT UNIT
//*       UNIT 9 CONTAINS THE LISTING OF THE CALCULATED
//*         OUTPUT VALUES
//GO.FT06F001 DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=7265),
// SPACE=(CYL,(7,3))
//GO.FT07F001 DD SYSOUT=B,DCB=(RECFM=FB,BLKSIZE=7280,LRECL=80)
//GO.FT09F001 DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=7265),
// SPACE=(CYL,(15,5))
//GO.SYSUDUMP DD SYSOUT=A
//GO.DATA5 DD *
.400      .405      .410      .415      .420
.425      .430      .435      .440      .445
.450      .460      .470      .480      .490
.500      .510      .520      .530      .540
.550      .560      .570      .580      .590
.600      .610      .620      .630      .640
.650      .660      .670      .680      .690
.700      .710      .720      .730      .740
.750      .760      .770      .780      .790
.800      .810      .820      .830      .840
.850      .860      .870      .880      .890
.900      .915      .930      .945      .960
.975      .990      1.10      1.12      1.14
1.16      1.18      1.20      1.22      1.24
1.26      1.28      1.30      1.32      1.34
1.36      1.38      1.40      1.44      1.48
1.50      1.52      1.54      1.56      1.58
1.60      1.62      1.64      1.66      1.68
1.70      1.72      1.74      1.76      1.78
1.80      1.84      1.88      1.92      1.96
2.00      2.04      2.08      2.12      2.16
2.20      2.24      2.28      2.32      2.36
2.40      2.44      2.48
.400      .405      .410      .415      .420
.425      .430      .435      .440      .445
.450      .460      .470      .480      .490
.500      .510      .520      .530      .540
```

.550	.560	.570	.580	.590
.600	.610	.620	.630	.640
.650	.660	.670	.680	.690
.700	.710	.720	.730	.740
.750	.760	.770	.780	.790
.800	.810	.820	.830	.840
.850	.860	.880	.900	.920
.940	.960	.980	1.00	1.02
1.04	1.06	1.08	1.10	1.12
1.14	1.16	1.18	1.20	1.22
1.24	1.26	1.28	1.30	1.32
1.34	1.36	1.38	1.40	1.44
1.48	1.50	1.52	1.54	1.56
1.58	1.60	1.62	1.64	1.66
1.68	1.70	1.72	1.74	1.76
1.78	1.80	1.84	1.88	1.92
1.96	2.00	2.04	2.08	2.12
2.16	2.20	2.24	2.28	2.32
2.36	2.40	2.44	2.48	

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
//ZBMJM001 JOB (S10122604D,T,L00360,002002),B16
//* THISDATE
// EXEC FORTRANH,PARM='OPT=2,XREF,ID,MAP,EBCDIC',REGION=300K
//* NOTE - DEFAULT REGION ON 75 AND 91 AFTER 6/12/74 IS ONLY 250K FOR
//* FORTRANH AND 150K FOR FORTRANG. MOST OF SKYLAB PROGRAMS USING
//* JCLFORT NEED 300K.
//SOURCE.SYSIN DD *
C       THIS PROGRAM WAS WRITTEN BY M. MACK OF C.S.C.
C       FOR TASK 092,SKYLAB,UNDER CONTRACT NAS-5-11999
C       IN  APRIL, 1974. IT USES A GAUSSIAN FILTER TO
C       'AVERAGE' SOLAR IRRADIANCE VALUES, SUPPLIED
C       ORIGINALLY BY DR. THEKAKARA THRU DR. CURRAN
C       (GSFC), AND CONVERTED TO COMPATIBLE UNITS
C       UNDER THE SAME TASK 092 BY PROGRAM 'CON'. THESE
C       'AVERAGES' ARE DERIVED CORRESPONDING TO EACH OF THE AVERAGES
C       OF THE WAVELENGTH LIMITS FOR ALL 20 ACTIVE SKYLAB S-192
C       CHANNELS.
C
      COMMON /DATATH/ WVTH(3297),AIRTH(3297)
      DIMENSION WAVSKY( 20),AIRSKY( 20),SIGMAF( 20)
C
C       LOGICAL VARIABLE USED IN CODING
C
      LOGICAL DO1
      DATA NOPTS,NPTS / 20,3297 /
C
C     NOPTS   IS THE NUMBER OF WAVELENGTHS ALTOGETHER
C             IN WAVSKY
C     NPTS    IS THE NUMBER OF WAVELENGTHS COVERED IN
C             DR. THEKAKARA'S DATA
C
      DATA SIGMAF / 0.02,.02,.025,.025,.025,.025,.04,.04,
     1 .05,.05,.1,.1,.125,.125,.05,.025,.025,.05,1.15,.025 /
C
C     SIGMAF IS THE FACTOR BY WHICH EACH WAVELENGTH
C             IS MULTIPLIED TO DETERMINE THE 'SIGMA'
C             VALUE FOR THE FILTER USED FOR THAT
C             WAVELENGTH
C       IT IS ACTUALLY EQUAL TO THE FILTER HALFWIDTH AT
C       HALF MAXIMUM,(I.E. TRANSMISSION IS ROUGHLY
C       EQUAL TO A*EXP(-X/SIGMA) WHERE A IS AN
C       ARBITRARY CONSTANT - SIGMAF WOULD BE THE
```

```
C        VALUE WHICH WOULD YIELD HALF THE
C        MAXIMUM TRANSMISSION VALUE)
C       (WHOLE WIDTHS,ACCORDING TO DR. CURRAN,ARE THE
C       DIFFERENCES IN THE LIMITS FOR EACH CHANNEL USED -
C       HERE, .04,.04,.05,.05,.05,.05,.08,.08,.1,.1,.2,.2,
C       .25,.25,1.,.05,.05,.1,2.3,.05 MICRONS)
C
C
C        FORMAT STATEMENTS
C
 1000 FORMAT(5F10.0)
 1500 FORMAT('THE FOLLOWING ARE THE ORDERED WAVELENGTHS',
     1' FOR SKYLAB'/,'DATA - USED IN CONJUNCTION WITH',
     2' CALCULATED-ASSOCIATED'/,'THEKAKARA DATA - S-192')
 1750 FORMAT(1H1, 'INPUT VALUES'/)
 2000 FORMAT(5X,'X',F7.3,',',F7.3,',',F7.3,',',F7.3,
     1 ',',F7.3,',')
 2500 FORMAT('END OF WAVELENGTH DATA - BEGINNING OF',
     1' CALCULATED-ASSOCIATED THEKAKARA'/,
     2'DATA - S192 COMPARISON')
 3000 FORMAT(5X,'X',F7.5,',',F7.5,',',F7.5,',',F7.5,
     1 ',',F7.5,',')
 3500 FORMAT(1H1,'DATA CALCULATED AND OUTPUT TO UNIT',
     1 ' 7'/,1X,3(7X,'WAVELENGTH',4X,'IRRADIANCE VALUE',
     2 5X)//)
 4000 FORMAT(1H ,I4,'.',2X,F10.6,8X,F9.7,8X,I4,'.',2X,
     1F10.6,8X,F9.7,8X,I4,'.',2X,F10.6,8X,F9.7)
 4500 FORMAT(1H0,'END OF PROGRAM - VALUES CALCULATED AND'
     1' PRINTED')
 6000 FORMAT(1H ,I5,4E16.8,3I5/,3X,5E16.8)
 7000 FORMAT(1H ,I5,F10.5,2I6,4E16.8)
C
C        AS THE FOLLOWING VALUE IS USED FREQUENTLY, BUT IS
C        A CONSTANT, AND AS WE WISH THE LIMIT OF MACHINE
C        ACCURACY,(AS WELL AS CONSISTENCY IN MACHINE
C        CALCULATION), WE CALCULATE IT HERE.
C
       ALN2=ALOG(2.)
C
C        WE NOW READ IN THE ARRAY OF DESIRED SKYLAB S-191
C        COMPARABLE WAVELENGTHS
C
       READ(5,1000) WAVSKY
C
C        THIS LOOP WILL CALCULATE THE 'AVERAGE' DESIRED FOR THE
```


REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
C       WAVELENGTHS CORRESPONDING TO THE AVERAGES OF THE
C       WAVELENGTH LIMITS FOR THE ACTIVE CHANNELS. OUTER
C       LOOP WILL DETERMINE LIMITS OF INNER LOOP
C
      DO 400 LB=1,NOPTS
      WAV=WAVSKY(LB)
      SIG=SIGMAF(LB)*WAV
C
C       HERE, THE 'SPREAD' AMONG THEKAKARA'S DATA FOR
C       NUMERICAL INTERPOLATION IS DETERMINED,(I.E.,HERE,
C       THE LIMITS OF THE INNER LOOP)
C
      W1=WAV-SIG
      W2=WAV+SIG
      DO1=.TRUE.
C
      DO 150 LZ=1,NPTS
      IF(.NOT. DO1) GO TO 100
      IF(W1.LT.WVTH(1)) GO TO 95
      IF(W1.GT. WVTH(LZ)) GO TO 150
      I1=LZ-1
      DO1=.FALSE.
      IF(I1.LT.1) I1=1
      GO TO 100
   95 CONTINUE
      I1=1
      DO1=.FALSE.
  100 CONTINUE
      IF(W2.GT.WVTH(LZ)) GO TO 150
      I2=LZ+1
      GO TO 200
  150 CONTINUE
C
  200 CONTINUE
      IF(I2.GT.NPTS) I2=NPTS
C
C       NOW THE 'AVERAGE' MAY BE CALCULATED(NUMERICAL
C       INTERPOLATION PERFORMED)
C
      SIGPRD =ALN2/(SIG*SIG)
      SUM1=0.
      SUM2=0.
      DO 300 LTR=I1,I2
C
C       FIRST CALSULATE WAVELENGTH DIFFERENCE
```

```
C
      WD=WVTH(LTR)-WAV
C
C
C       NOW CALCULATE -((NU-NUO)/SIGMA)**2 * LN2
C
      WD2P=-WD*WD*SIGPRD
C
C       TAKE EXPONENTIAL OF ABOVE
C
      WEXP=EXP(WD2P)
      SUM1=SUM1+WEXP
C
C       TAKE PRODUCT OF EXPONENTIAL VALUE AND IRRADIANCE
      ARWEXP=AIRTH(LTR)*WEXP
      SUM2=SUM2+ARWEXP
      WRITE(6,6000) LB,WAV,W1,W2,SIG,LTR,I1,I2,
     1WD,WD2P,SUM1,ARWEXP,SUM2
C
  300 CONTINUE
      AIRSKY(LB)=SUM2/SUM1
  400 CONTINUE
C
C
C       INPUT VALUES ARE LISTED FOR CHECKING PURPOSES
C
      WRITE(9,1750)
      WRITE(9,4000) (LX,WVTH(LX),AIRTH(LX),LX=1,NPTS)
C
C       WE NEXT PRINT OUT RESULTS BOTH ON
C       CARDS ,(FOR FUTURE USE), AND ON UNIT '9',(SYSOUT A),
C       FOR SCIENTISTS TO STUDY.
C
C         THIS LOOP WAS INSERTED TO CREATE SEVERAL DECKS
C           AT ONCE WITHOUT WAITING FOR IBM 20 TURN-AROUND.
      DO 900 LDECK=1,4
      WRITE(7,1500)
      WRITE(7,2000)WAVSKY
      WRITE(7,2500)
      WRITE(7,3000) AIRSKY
C
C         END OF DECK LOOP
  900 CONTINUE
      WRITE(9,3500)
      WRITE(9,4000) (LD,WAVSKY(LD),AIRSKY(LD),LD=1,NOPTS)
```


REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

```
      WRITE(6,4500)
      RETURN
      END
// EXEC LINKGO,REGION.GO=200K
//LINK.OBJECT DD *
 INCLUDE LOADLIB(ZBMJMCRB)
//*    UNIT 6 IS THE DEBUG AND GENERAL MESSAGE UNIT ON PRINTER
//*     UNIT 7 IS THE CARD OUTPUT UNIT
//*      UNIT 9 CONTAINS THE LISTING OF THE CALCULATED
//*        OUTPUT VALUES
//GO.FT06F001 DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=7265),
// SPACE=(CYL,(7,3))
//GO.FT07F001 DD SYSOUT=B,DCB=(RECFM=FB,BLKSIZE=7280,LRECL=80)
//GO.FT09F001 DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=7265),
// SPACE=(CYL,(15,5))
//GO.SYSUDUMP DD SYSOUT=A
//GO.DATA5 DD *
.54       .54       .585      .585      .645
.645      .72       .72       .83       .83
1.65      1.65      2.225     2.225     12.5
.485      1.005     1.14      11.35     .435
// EXEC NOTIFY,MODE=ALL,MSG='JOB IS COMPLETE'
// EXEC LINKGO,REGION.GO=200K
//LINK.SYSLIN DD *
 INCLUDE LOADLIB(ZBMJML91)
//*    UNIT 6 IS THE DEBUG AND GENERAL MESSAGE UNIT ON PRINTER
//*     UNIT 9 CONTAINS THE LISTING OF THE TAPE CONTENTS
//*     TPS191 IS THE INPUT DATA TAPE
//GO.FT06F001 DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=7265),
// SPACE=(CYL,(7,3))
//GO.FT09F001 DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=7265),
// SPACE=(CYL,(15,5))
//GO.TPS191 DD UNIT=2400-9,DCB=(RECFM=U,BLKSIZE=2520),LABEL=(,BLP),
// VOL=SER=L0019
//GO.SYSUDUMP DD SYSOUT=A
//GO.DATA5 DD *
// EXEC LINKGO,REGION.GO=300K
//LINK.SYSLIN DD *
 INCLUDE LOADLIB(ZBMJMT91)
//*    UNIT 6 IS THE DEBUG AND GENERAL MESSAGE UNIT ON PRINTER
//*     UNIT 9 CONTAINS THE LISTING OF THE NEW TAPE'S CONTENTS
//*       (IT IS DEFINED AS A PRINTER UNIT ALSO)
//*      UNIT 12 IS THE OUTPUT UNIT FOR THE NEW TAPE
//*     TP191A IS THE INPUT DATA TAPE OF FORM'A'
//*     TP191B IS THE INPUT DATA TAPE OF FORM'B'
```

```
//GO.FT06F001 DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=7265),
// SPACE=(CYL,(10,5))
//GO.FT09F001 DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=7265),
// SPACE=(CYL,(20,5))
//GO.FT12F001 DD UNIT=(2400-9,,DEFER),DCB=(RECFM=VBS,
// LRECL=1796,BLKSIZE=30536,DEN=3),LABEL=(,BLP),
// VOL=SER=DUMMY
//GO.TP191A DD UNIT=2400-9,DCB=(RECFM=U,BLKSIZE=2520,EROPT=ACC),
// LABEL=(,BLP),VOL=SER=L0001
//GO.TP191B DD UNIT=2400-9,DCB=(RECFM=U,BLKSIZE=2520,EROPT=ACC),
// LABEL=(,BLP),VOL=SER=L0002
//GO.SYSUDUMP DD SYSOUT=A
//GO.DATA5 DD *
// EXEC LINKGO,REGION.GO=200K
//*    UNIT 6 IS THE DEBUG AND GENERAL MESSAGE UNIT ON PRINTER
//*     UNIT 9 CONTAINS THE LISTING OF THE TAPE CONTENTS
//*     TPS191 IS THE INPUT DATA TAPE
//GO.FT06F001 DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=7265),
// SPACE=(CYL,(7,3))
//GO.FT09F001 DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=7265),
// SPACE=(CYL,(15,5))
//GO.TPS191 DD UNIT=2400-9,DCB=(RECFM=U,BLKSIZE=2520),LABEL=(,BLP),
// VOL=SER=L0001
//GO.SYSUDUMP DD SYSOUT=A
//GO.DATA5 DD *
// EXEC LINKGO,REGION.GO=200K
//*    UNIT 6 IS THE DEBUG AND GENERAL MESSAGE UNIT ON PRINTER
//*     UNIT 9 CONTAINS THE LISTING OF THE NEW TAPE'S CONTENTS
//*       (IT IS DEFINED AS A PRINTER UNIT ALSO)
//*      UNIT 12 IS THE OUTPUT UNIT FOR THE NEW TAPE
//*     TP191A IS THE INPUT DATA TAPE OF FORM'A'
//*     TP191B IS THE INPUT DATA TAPE OF FORM'B'
//GO.FT06F001 DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=7265),
// SPACE=(CYL,(7,3))
//GO.FT09F001 DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=7265),
// SPACE=(CYL,(15,5))
//GO.FT12F001 DD UNIT=(2400-9,,DEFER),DCB=(RECFM=VBS,
// LRECL=1452,BLKSIZE=29044,DEN=3),LABEL=(,BLP),
// VOL=SER=DUMMY
//GO.TP191A DD UNIT=2400-9,DCB=(RECFM=U,BLKSIZE=2520),LABEL=(,BLP),
// VOL=SER=L0021
//GO.TP191B DD UNIT=2400-9,DCB=(RECFM=U,BLKSIZE=2520),LABEL=(,BLP),
// VOL=SER=L0022
//GO.SYSUDUMP DD SYSOUT=A
//GO.DATA5 DD *
```

```
// EXEC LINKGO,REGION.GO=220K
//LINK.OBJECT DD *
 INCLUDE LOADLIB(ZBMJMR92)
//* LOADLIB ZBMJMR92 CONTAINS READ PROGRAMS FOR S-192
//*     UNIT 6 IS THE DEBUG AND GENERAL MESSAGE UNIT ON PRINTER
//*      UNIT 9 CONTAINS THE LISTING OF THE TAPE CONTENTS
//*       TP192 IS THE INPUT DATA TAPE
//GO.FT06F001 DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=7265),
// SPACE=(CYL,(7,3))
//GO.FT09F001 DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=7265),
// SPACE=(CYL,(15,5))
//GO.TP192 DD UNIT=2400-9,DCB=(RECFM=U,BLKSIZE=32000),LABEL=(,BLP),
// VOL=SER=L0045
//GO.SYSUDUMP DD SYSOUT=A
//GO.DATA5 DD *
```

APPENDIX F - TEMPERATURE HUMIDITY PROFILE PROGRAM PROGRAM LISTINGS, INPUT, AND JCL OUTPUT

```
//ZBMJM001 JOB (S10122604D,T,L00360,002002),B16
//* THISDATE
// EXEC FORTRANH,PARM='OPT=2,XREF,ID,MAP,EBCDIC',REGION=300K
//* NOTE - DEFAULT REGION ON 75 AND 91 AFTER 6/12/74 IS ONLY 250K FOR
//* FORTRANH AND 150K FOR FORTRANG. MOST OF SKYLAB PROGRAMS USING
//* JCLFORT NEED 300K.
//SOURCE.SYSIN DD *
C
C       THIS PROGRAM WAS WRITTEN BY M.MACK OF CSC FOR TASK 092,
C       SKYLAB, UNDER CONTRACT NAS 5-11999 IN MARCH,1974.
C       IT IS THE TEMPERATURE-HUMIDITY PROFILE PROGRAM.
C       S-191 DATA,(IN ITS GSFC FORM), IS READ AND TESTS
C       ARE PERFORMED TO DETERMINE THE PRESENCE OR NON-PRESENCE
C       OF CLOUDS. THE RESULTS DETERMINE WHETHER OR NOT
C       SUBROUTINES CLOUD OR NCLOUD ARE CALLED.
C
C        THE INPUT DATA ARRAY IS CALLED DATUS. IT IS EQUIVALENCED
C        TO THREE OTHER ARRAYS:IDATUS,WAVEL AND RADVAL.IDATUS IS
C        AN INTEGER*2 ARRAY 6 HALF-WORDS LONG,WHICH WILL CONTAIN
C        THE TIME OF THE SPECTRUM.WAVEL IS THE ARRAY OF WAVELENGTH
C        VALUES FOR S-191 AND RADVAL IS THE ARRAY OF ASSOCIATED
C        RADIANCE VALUES.
C        DATUS IS ALSO EQUIVALENCED TO SINGLE VARIABLES:ZENITH TO
C          THE 10TH ENTRY.
C
C
C        TAPEN CONTAINS THE NAME OF THE TAPE TO BE MOUNTED,(VOLUME
C        AND SERIAL NUMBER)
C
C        CONE AND CTWO ARE THE CONSTANTS USED IN PLANCK'S EQUATION
C        TO DETERMINE THE EQUIVALENT BRIGHTNESS TEMPERATURE.
C        PI IS THE MATHEMATICAL CONSTANT OF THE SAME NAME.
C        CNDGRD IS A CONVERSION FACTOR USED TO CHANGE ANGLES IN
C        DEGREES TO ANGLES IN RADIANS.
C        REFLLM IS THE DESIRED LIMIT ON REFLECTANCE BELOW WHICH,(FOR
C        .74 MICRONS) THERE IS PROBABLY NO CLOUD.
C        TEMPLM IS THE DESIRED LIMIT ON TEMPERATURE BELOW WHICH,(FOR
C        THE TEMPERATURE ASSOCIATED WITH 11.1 MICRONS) THERE IS PROBABLY
C        A CLOUD.
C        THEKVL IS THEKAKARA'S IRRADIANCE VALUE AT .74 MICRONS USED IN
C        CALCULATING REFLECTANCE.
C        WV148 WILL BE THE WAVELENGTH VALUE OF 11.1 MICRONS.
C        WV1485 WILL BE THE FIFTH POWER OF 11.1 MICRONS,(THIS
C        AND WV148 ARE USED IN CALCULATING BRIGHTNESS TEMPERATURE)
C
```


REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
C       . CLOUD1 AND CLOUD2 ARE LOGICAL VARIABLES CORRESPONDING TO
C           THE RESULTS OF THE TWO TESTS.IF A CLOUD IS SUSPECTED, THE
C           VARIABLE IS SET TO TRUE. IF NOT, IT IS SET TO FALSE.
C
C
C         READ1 IS A LOGICAL VARIABLE WHICH SHOWS IF THIS IS THE
C           FIRST TAPE REQUESTED. IF IT IS NOT, THE OLD ONE MUST BE
C           UNLOADED
C
C         THE FOLLOWING ARE USED FOR STATISTICS PURPOSES:
C          NSPEC WILL BE THE NUMBER OF SPECTRA READ
C          NCLD WILL BE THE NUMBER CONSIDERED TO HAVE CLOUDS IN THE FIELD
C           OF VIEW.
C          NNCLD WILL BE THE NUMBER CONSIDERED TO HAVE NO CLOUDS IN THE
C           FIELD OF VIEW.
C          NDUB WILL  BE THE NUMBER WHICH WERE INCONCLUSIVELY
C           CLOUD SPECTRA,(I.E. TESTS WERE NOT CONSISTENT SO THEY WERE
C           ARBITRARILY CALLED CLOUD SPECTRA).
C          NTAPE WILL CONTAIN THE NUMBER OF TAPES REQUESTED AND UTILIZED
C          NERR WILL BE THE NUMBER OF READ ERRORS ENCOUNTERED IN ACCESSING
C           THE TAPES.
C
C        NAMELIST,INPUT,WILL CONTAIN THE VARIABLES USED TO
C        INITIALIZE THE WOLF PLOT PACKAGE
C
C
C        AT USER OPTION, TIME PERIODS OF A TAPE OR TAPES MAY BE
C        SCANNED,(RATHER THAN MERELY THE ENTIRE TAPE OR TAPES).
C
      DIMENSION DATUS(448),WAVEL(217),RADVAL(217)
      INTEGER*2 IDATUS(896)
      DIMENSION        IDEV(3),IVALE(5)
      REAL*8 NLST /' &INPUT '/
      REAL*8 PLOTER,PLOTS(5)
      REAL*8 TAPEN,WV1485
      LOGICAL CLOUD1/.TRUE./
      LOGICAL CLOUD2/.FALSE./
      LOGICAL FIRSTM /.FALSE. /
      LOGICAL READ1/.TRUE./
      EQUIVALENCE (DATUS(1),IDATUS(1)),(DATUS(15),WAVEL(1)),
     1(DATUS(10),ZENITH),(DATUS(232),RADVAL(1))
      NAMELIST /INPUT/ PLOTER,IPRINT,MONTH,IDAY,IYEAR,IHOUR,
     1 MINUTE,ISEC,IENDHR,IENDMI,IENDSC
      DATA PLOTS /'CALCOMP ','SC4020FH','SC40208H','SC4020CF',
     1 'PRINTER '/
```

```
      DATA IVALE /2000,0030,0020,0010,0001/
      DATA IPRINT /0/
      DATA MONTH /0/
      DATA NUMM /4/
      DATA NTIMOL /0/
      DATA CONE /3.7413E+04/
      DATA CTWO /1.4388E+04/
      DATA PI /3.1415 /
      DATA CNDGRD /.017453 /
      DATA REFLLM / .20 /
      DATA TEMPLM / 273. /
      DATA THEKVL /.12602 /
      DATA WV165 / 11.1 /
      DATA WV1655 / 1.685058155D+05/
      DATA NSPEC,NCLD,NNCLD,NDUB,NTAPE,NERR/0,0,0,0,0,0 /
      DATA NO1,NO2,NO3,NO4,NO5,NOGO/207,9000,5,2,210,1/
C
C       FORMAT STATEMENTS
C
 1000 FORMAT(A8)
 2000 FORMAT(1HO,'DUBIOUS CASE NO. ',I5/,6X,'TIME WAS',1X,3I3,
     11X,3I3,2X,'SUN ZENITH ANGLE WAS',1X,F7.3/,12X,
     2'RADIANCE VALUE FOR .74 MICRONS WAS ',E15.8,1X,
     3'YIELDING A REFLECTANCE VALUE OF ',F8.4,1X,
     4'COMPARED TO A LIMIT OF ',F6.3/,12X,
     5'RADIANCE VALUE FOR 11.1 MICRONS WAS ',E15.8,1X,
     6'YIELDING A TEMPERATURE VALUE OF ',F8.3/,14X,
     7'COMPARED TO A LIMIT OF ',F8.3//)
 2100 FORMAT(1HO,'ERROR',I5,' IN READING OF TAPE ',A8/)
 2200 FORMAT(1HO,'END OF REQUESTED TIME PERIOD:',3I3,1X,3I3,
     1 ' TO ',3I3,1X,3I3/,1X,5X,'THERE WERE ',I5,1X,
     2 'SPECTRA READ WITH ',I5,1X,'SPECTRA TESTED.'/,1X,
     3 6X,'OF THESE,',I5,1X,'WERE CONSIDERED CLOUD SPECTRA,',
     4 'WHILE ',I5,1X,'WERE CONSIDERED NO-CLOUD SPECTRA -'/,8X,
     5 ' (THE CLOUD SPECTRA NUMBER INCLUDES',1X,I5,1X,'WHICH ',
     6 'WERE INCONCLUSIVELY CLOUD SPECTRA.)'//)
 2300 FORMAT(1HO,'END OF FILE ENCOUNTERED ON TAPE',1X,A8,
     12X,'DETERMINE IF ANOTHER TAPE IS DESIRED'/)
 2400 FORMAT(1HO,'NO ADDITIONAL TAPES WERE REQUESTED - THERE ',
     2'WERE A TOTAL OF ',I5,1X,'TAPES MOUNTED FOR THIS JOB'/,
     310X,'THERE WERE ',I10,1X,'SPECTRA TESTED. OF THESE ',
     4I5,1X,' WERE CONSIDERED CLOUD SPECTRA, WHILE',1X,I5,1X,
     5'WERE CONSIDERED'/,19X,'NO-CLOUD SPECTRA  -  (THE CLOUD ',
     6'SPECTRA NUMBER INCLUDES',I5,1X,'WHICH WERE INCONCLUSIVELY ',
     7'CLOUD SPECTRA.)'//)
```

```
C
C *****
C
C       DEBUG AID
C
        CALL ERRSET(NO1,NO2,NO3,NO4, NOGO,NO5)
C
        ICASE=0
        WRITE(6,3010)
 3010 FORMAT(1H1, 'DEBUG DATA'/)
C
C
C *****
        READ(5,INPUT)
        REWIND 5
        DO 5 L=1,5
        NUM1=L
        IF(PLOTER.EQ.PLOTS(L)) GO TO 6
    5   CONTINUE
    6   CONTINUE
        NUM=IVALE(NUM1)
        NUM=NUM+IPRINT
        CALL NOSTAE
        CALL CPRIME(2)
        CALL PLOTST(NUM,NUMM)
   10   CONTINUE
C
C       NEXT DESIRED TAPE,(VOLUME AND SERIAL NO.) IS READ
C
        READ(5,1000,END=999) TAPEN
        IF(TAPEN.EQ.NLST) GO TO 999
        NTAPE=NTAPE+1
        IF(READ1) GO TO 20
C
C       IF THERE IS A TAPE ON THE DRIVE ALREADY,( I.E. THIS IS
C       NOT THE FIRST TAPE REQUESTED), UNLOAD THE OLD ONE.
C
        CALL UNLOAD(12)
   20   CONTINUE
        READ1=.FALSE.
C
C       MOUNT THE NEW TAPE
C
        CALL MOUNT(1,12,TAPEN)
   30   CONTINUE
```


REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
C
C       NEXT DATA SET IS READ
C
        CALL FREAD(IDATUS,12,LENGTH,&99,&70)
        NSPEC=NSPEC+1
        IF(MONTH.EQ.0) GO TO 35
        IF(FIRSTM) GO TO 36
        IF(IDATUS(3).LT.IYEAR) GO TO 10
        IF(IDATUS(1).LT.MONTH) GO TO 10
        IF((IDATUS(1).EQ.MONTH).AND.(IDATUS(2).LT.IDAY)) GO TO 10
        IF((IDATUS(1).EQ.MONTH).AND.(IDATUS(2).EQ.IDAY).AND.
     1   (IDATUS(4).LT.IHOUR)) GO TO 10
        IF((IDATUS(1).EQ.MONTH).AND.(IDATUS(2).EQ.IDAY).AND.
     1   (IDATUS(4).EQ.IHOUR).AND.(IDATUS(5).LT.MINUTE)) GO
     2     TO 10
        IF((IDATUS(1).EQ.MONTH).AND.(IDATUS(2).EQ.IDAY).AND.
     1   (IDATUS(4).EQ.IHOUR).AND.(IDATUS(5).EQ.MINUTE).AND.
     2     (IDATUS(6).LT.ISEC)) GO TO 10
        FIRSTM=.TRUE.
  36    CONTINUE
        IF(IDATUS(4).GT. IENDHR) GO TO 80
        IF((IDATUS(4).EQ.IENDHR).AND.(IDATUS(5).GT.IENDMI))GO TO 80
        IF((IDATUS(4).EQ.IENDHR).AND.(IDATUS(5).EQ.IENDMI).AND.
     1   (IDATUS(6).GT.IENDSC)) GO TO 80
        NTIMOL=NTIMOL+1
  35    CONTINUE
C
C        TEST 1 DETERMINES IF THE REFLECTANCE AT .74 MICRONS IS LESS THAN
C          THE GIVEN LIMIT
C
        CLOUD1=.TRUE.
        ZENAN=ZENITH*CNDGRD
        COSZN=COS(ZENAN)
        IF(COSZN.LE.0.) GO TO 40
        REFLEC=(RADVAL(40)*PI)/(COSZN*THEKVL)
        IF(REFLEC.LT.REFLLM) CLOUD1=.FALSE.
  40    CONTINUE
C
C        TEST 2 DETERMINES IF THE TEMPERATURE AT 11.1 MICRONS IS
C          LESS THAN THE INPUT LIMIT. IF SO,A CLOUD IS PRESENT
C
        CLOUD2=.TRUE.
        IF(RADVAL(165).LE.0.) GO TO 60
        ALOGRG=1. + (CONE/(PI*WV1655*RADVAL(165)))
        IF (ALOGRG.LE.0.) GO TO 60
```

```
      DENOM=WV165*ALOG(ALOGRG)
      IF(DENOM.LE.0.) GO TO 60
      TEMPVL=CTWO/DENOM
      IF(TEMPVL.GE.TEMPLM) CLOUD2=.FALSE.
   60 CONTINUE
C
C       IF BOTH TESTS SHOWED CLOUDS WERE PRESENT,(CLOUD1 AND
C         CLOUD2 WERE BOTH TRUE), THEN IT IS ASSUMED A CLOUD
C        IS PRESENT. IF BOTH TESTS SHOWED NO CLOUD WAS PRESENT,
C        IT IS ASSUMED NO CLOUD WAS PRESENT. IF TESTS GAVE OPPOSITE
C        RESULTS, IT IS ASSUMED A CLOUD WAS PRESENT.
C
      IF((.NOT.CLOUD1).AND.CLOUD2) NDUB=NDUB+1
      IF(CLOUD1 .AND. (.NOT. CLOUD2)) NDUB=NDUB+1
      IF((.NOT. CLOUD1) .AND. CLOUD2) WRITE(6,2000) NDUB,IDATUS,
     1ZENITH,RADVAL(40),REFLEC,REFLLM,RADVAL(165),TEMPVL,TEMPLM
      IF(CLOUD1 .AND. (.NOT. CLOUD2)) WRITE(6,2000)NDUB,IDATUS,
     1ZENITH,RADVAL(40),REFLEC,REFLLM,RADVAL(165),TEMPVL,TEMPLM
      IF(CLOUD1 .OR. CLOUD2 ) NCLD=NCLD+1
      IF(CLOUD1 .OR. CLOUD2) CALL CLOUD(DATUS,IDATUS)
      IF((.NOT.CLOUD1) .AND. (.NOT. CLOUD2) ) NNCLD=NNCLD+1
      IF((.NOT.CLOUD1) .AND. (.NOT. CLOUD2)) CALL NCLOUD(DATUS,IDATUS)
C *****
C
C       DEBUG AID
C
      ICASE=ICASE+1
      WRITE(6,3000) ICASE,(IDATUS(L),L=1,6),ZENITH,ZENAN,COSZN,
     1 RADVAL(40),REFLEC,REFLLM,CLOUD1,RADVAL(165),ALOGRG,DENOM,
     2 TEMPVL,TEMPLM,CLOUD2
 3000 FORMAT(1H0,I5,'.',1X,'TIME=',3I3,1X,3I3,2X,'ZENITH=',F7.3,
     1 2X,'OR',F8.5,2X,'COS(ZENITH)=',F8.5/,9X,'RAD40=',F9.6,2X,
     2 'REFLEC=',F9.6,2X,'REFLLM=',F9.6,3X,'CLOUD1=',L4/,9X,
     3 'RAD165=',F9.6,2X,'ALOGRG=',E16.8,2X,'DENOM=',E16.8,2X,
     4 'TEMPVL=',F8.3,2X,'TEMPLM=',F8.3,3X,'CLOUD2=',L4/)
C
C *****
      GO TO 30
   70 CONTINUE
C
C       ERROR IN READING TAPE WAS ENCOUNTERED
C
      NERR=NERR+1
      WRITE(6,2100) NERR,TAPEN
      GO TO 30
```


REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
   80  CONTINUE
C
C       END OF REQUESTED TIME PERIOD
C
       WRITE(6,2200) MONTH,IDAY,IYEAR,IHOUR,MINUTE,ISEC,MONTH,
      1 IDAY,IYEAR,IENDHR,IENDMI,IENDSC,NSPEC,NTIMOL,NCLD,
      2 NNCLD,NDUB
       GO TO 9999
   99  CONTINUE
C
C       END OF FILE ENCOUNTERED ON CURRENT TAPE - DETERMINE IF NEW
C        TAPE IS DESIRED
C
       WRITE(6,2300) TAPEN
       GO TO 10
  999 CONTINUE
C
C       NO NEW TAPES WERE REQUESTED - PROGRAM CONCLUDES WITH
C         STATISTICS LISTED
C
       CALL UNLOAD(12)
       WRITE(6,2400) NTAPE,NSPEC,NCLD,NNCLD,NDUB
 9999 CONTINUE
       CALL SETBLK(999)
       CALL ENDPLT
       RETURN
       END
       SUBROUTINE CLOUD(DATUS,IDATUS)
       DIMENSION DATUS(448),IDATUS(448)
C       DUMMY SUBROUTINE CALL
       RETURN
       END
       SUBROUTINE DTAUDU(T,Q,M,N,DH,PT,VWCOR)
C *** GSFC PROGRAM NO. S00091   NIMBUS-4 TEMP./HUMIDITY ESTIMATION PROG
C
       COMMON /H2O/ TO( 99),QO( 99)
       COMMON/H2O/ TAU(28,100),C0(28),C1(28),C2(28),C3(28),C4(28),C5(28),
      # C6(28),C7(28),C8(28)
       DIMENSION T(1),Q(1),     PP(100)
       LOGICAL FSTIME/.TRUE./
C
C
       IF(.NOT.FSTIME)GO TO 110
       FSTIME = .FALSE.
       APT = ALOG(PT)
```

```
      DO 100 J=1,N
      H = APT + (J-1)*DH
100   PP(J) = EXP(H)
110   PAVOLD = 0.
      TAVOLD = 0.
      POLD = PT
      UOLD = 0.
      DO 1 J=2,N
      P = PP(J)
      IF(O(J-1).EQ.0.) GO TO 2
C
C
C       U IS THE INTEGRATED MASS
C       O IN THESE EQUATIONS CORRESPONDS TO THE W IN PAPER BY DR. CONRATH
C       980 IS GRAVITY IN CM/SEC**2
C
      U = UOLD + O(J-1)*(P-POLD)/980.
      PAV = (UOLD*PAVOLD + .5*(P + POLD)*(U - UOLD))/U
      TAV = (UOLD*TAVOLD + T(J - 1)*(U - UOLD))/U
      GO TO 3
    2 CONTINUE
      DO 4 I=1,M
    4 TAU(I,J) = 0.
      POLD = P
      GO TO 1
    3 CONTINUE
      X = ALOG(U*273./TAV)
      Y = ALOG(PAV/1000.)
      Z = ALOG(TAV/273.)
      PAVOLD = PAV
      TAVOLD = TAV
      UOLD = U
C
C       22 HERE IS THE DISPLACEMENT NEEDED TO PUT INDEX AT HUMIDITY
C       MIXING RATIO VALUES.(THERE ARE 22 TEMPERATURE CORRESPONDING
C       VALUES PRECEDING THEM)
C
C       TAU IS THE ATMOSPHERIC TRANSMITTANCE
C
C
      DO 1 K=1,M
      I = K+22
      IF(CO(I).NE.0)GO TO 10
      TAU(K,J) = 1.
      GO TO 1
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
10      W = C0(I) + C1(I)*X + C2(I)*Y + C3(I)*Z + C4(I)*X*Y*.1 + C5(I)*X
     # *Z + C6(I)*(X*X)*.1 +C7(I)*(X*X)*Z*.1 + C8(I)*Y*Z*Z*.1
        IF((W.LT.-10.).OR.(W.GT.2.5)) GO TO 7
        WX = C1(I) + .1*C4(I)*Y + C5(I)*Z + 2.*C6(I)*X*.1 + 2.*C7(I)*X*Z*
     1.1
        XU = 1./U
        WU = WX*XU
C
C         VWCOR IS THE SECANT OF THE NADIR VIEW ANGLE
C         - IT IS USED TO MODIFY DTAUDU VALUES,(STORED
C         HERE IN TAU).
C
        TAU(K,J) = -EXP(-EXP(W)*VWCOR)*(EXP(W)*VWCOR)*WU
        GO TO 5
      7 TAU(K,J) = 0.
      5 CONTINUE
        POLD = P
      1 CONTINUE
        DO 6 I=1,M
      6 TAU(I,1) = 0.
        RETURN
        END
        SUBROUTINE ESTH2O(XIM,XLAM,M,UTOTAL,T,TB,II,RR,CVGH2O,VWCOR,KTYPE)
C *** GSFC PROGRAM NO. S00091   NIMBUS-4 TEMP./HUMIDITY ESTIMATION PROG
        COMMON /H2O/ TO( 99),QO( 99),TAU(28,100)
        COMMON/CO2/TAUCO2(28,100)
        COMMON/CHA/                        RMSTMP,RMS
        DIMENSION T(1),XIM(1),XIO( 6),XIXIO( 6),XLAM(1),XI( 6),XICD( 6)
        DIMENSION DELS( 6),XK(100),    S( 6, 6),RH(100),ES(100)
        REAL*8 S
        LOGICAL HOTGRD
        COMMON/XXX/  TS, PS, PT, DH, N, NL,                              R,
       #                                    MTOTAL,
       # P( 99),T1( 99),
       #                              HOTGRD
        LOGICAL                       HOTGRD
C
C         FPWM IS THE MONONUCLEATION FREEZING POINT OF WATER
C
        DATA FPWM/234./
C
C
C         NOTE - NN IS 23 INSTEAD OF 100 OR 99 SINCE THE UPPER PORTIONS
C         OF THE ATMOSPHERE CONTRIBUTE NOTHING,(OR VERY LITTLE), TO THE
C         WATER VAPOR QUANTITIES - ONLY THE LOWER LEVELS,(DR. CONRATH
```

```
C       ESTIMATES THE LOWER 23). HAVE ANY EFFECTIVE CONTRIBUTION TO
C       MAKE.
C
        DATA C,MAX,NN/622.0, 5, 23/
        DATA C1,C2 /3.7413E+04,1.4388E+04/
        DATA PI/3.14159/
C
C       VP1 IS THE VAPOR PRESSURE EQUATION FOR ICE
C       VP2 IS THE VAPOR PRESSURE EQUATION FOR WATER
C
C
        VP1(X) = 6.11*EXP(22.49 - (6141./X))
        VP2(X) = 6.11*EXP(19.84 - (5417./X))
        FSB(X,Y)=C1*(X**(-5))/(PI*(EXP(C2/(X*Y))-1.))
C
C
        OLDRMS = 99999.
C
        IF(HOTGRD) GO TO 40
        DO 30 I=1,M
        XI(I)=XIM(I)
   30   CONTINUE
40      CONTINUE
C
        DO 115 I=1,NL
        IF (T(I) .LT. FPWM) GO TO 116
        ES(I) = VP2(T(I))
        GO TO 217
116     ES(I) = VP1(T(I))
C
C       RH IS THE RELATIVE HUMIDITY ARRAY
C
217     RH(I) = 0.5
        IF(P(I).LT.100)GO TO 117
        QQ(I) = (622.0 * 0.5 * ES(I)) / P(I)
        GO TO 115
C
C       AT THE TROPOPAUSE AND ABOVE,(100MB AND LOWER),THE MIXING RATIO
C       IS A CONSTANT OF ABOUT .001
C       THERE IS A COLD TRAP THERE
C
117     QQ(I) = 0.001
115     CONTINUE
C       BEGIN ITERATIVE LOOP
        DO 100 L=1,MAX
```


REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
C       WRITE(6,66) L
C   66 FORMAT (1H0,61X,'ITERATION ',I3)
C       CHECK RESIDUALS
C
C        THE ARGUMENT,KTYPE, IN EACH OF THESE SUBROUTINES IS THE
C        DISPLACEMENT NEEDED TO BYPASS TEMPERATURE PORTIONS OF ARRAYS.
C
        CALL TRANSW(T,OD,M,N,DH,PT,KTYPE,VWCOR)
        CALL TRANSC(T,M,N,DH,PT,KTYPE,VWCOR)
3000    CONTINUE
        DO 2 J=1,N
        DO 2 I=1,M
      2 TAU(I,J) = TAU(I,J)*TAUCO2(I,J)
C  *** HOT GROUND CORRECTION
        IF(.NOT.HOTGRD)GO TO 15
        DO 10 I=1,M
        XI(I)=XIM(I)-(FSB(XLAM(I),TB)-FSB(XLAM(I),TS))*TAU(I,N)
   10   CONTINUE
15      CONTINUE
        CALL RADH2O(T,XLAM,T(NL),M,N,XIO)
C
C***  CORRECT RADIANCES IF FIRST ITERATION
        IF(L.GT.1) GO TO 90
        CALL RADH2O(T,XLAM,T(II),M,II,XICD)
        DO 80 I=1,M
80      XI(I) = XIM(I) + XN * (XIO(I) - XICD(I))
C
90      CONTINUE
        RMS = 0.
        DO 101 I=1,M
        DEL = XI(I) - XIO(I)
        XIXIO(I) = DEL
C       WRITE(6,151) DEL
C 151 FORMAT(60X,E13.4)
101     RMS = RMS + DEL*DEL
        RMS = (RMS/M)**.5
C       WRITE(6,152) RMS
C 152 FORMAT(1H0,55X,'RMS RESIDUAL = ',E10.4)
C       CALCULATE REVISED H2O ESTIMATE
        CALL DTAUDU(T,OD,M,N,DH,PT,VWCOR)
C       DTAU/DU IS NOW STORED IN TAU(I,J)
        DO 105 I=1,M
        XK(1)=TAU(I,N)*TAUCO2(I,N) * .5 *(FSB(XLAM(I),T(N-1)) -
       1  FSB(XLAM(I),T(N-2)))
C
```

```
C       SECOND HALF OF ABOVE EQUATION CORRESPONDS TO THE 1/2 CHANGE IN
C       B(DELTA B) IN EQUATION 10 FROM PAPER BY DR. CONRATH ET AL.
C
        LN = NN - 1
        DO 106 J=1,LN
        XK(J+1)=XK(J) + TAU(I,N-J)*TAUCO2(I,N-J) *
       1 (FSB(XLAM(I),T(N-J)) - FSB(XLAM(I),T(NL-J)))
  106 CONTINUE
        DO 105 J=1,NN
C
C       ((C/980.)*DH) CAN BE TRACED TO EQUATIONS,(NEAR EQ. 7), IN
C       PAPER BY DR. CONRATH,(I.E. THE UNEXPRESSED BUT PRESENT BACKGROUND
C       EQUATIONS)
C
  105 TAU(I,J) = XK(  J) *ES(N-J) * ((C/980.)*DH)
C
C       THE KERNEL MATRIX IS NOW STORED IN TAU
C
C
C
C       EQUATIONS IN HERE ROUGHLY CALCULATE THE SUMMATION
C       OF THE PRODUCT OF THE SUMMATION OF THE TRANSPOSE OF K TIMES THE
C       INVERSE OF THE PRODUCT OF K AND THE TRANSPOSE OF K  TIMES
C       XIXIO,(WHERE K IS THE KERNEL MATRIX)
C
        DO 107 I=1,M
        DO 107 K=1,M
        SUM = 0.
        DO 108 J=1,NN
  108 SUM = SUM + TAU(I,J)*TAU(K,J)
  107  S(I,K) = SUM
        DO 109 I=1,M
C
C       RR IS THE DAMPING FACTOR WHICH ON NIMBUS WAS SUPPLIED BY I.REVAH
C       THIS DAMPING FACTOR CORRRSPONDS TO THE GAMMA IN THE PAPER .
C
109     S(I,I) = S(I,I) + RR
        CALL MATINH(S,M,0)
        DO 112 I=1,NN
        SIM = 0.
        DO 113 J=1,M
        SOM = 0.
        DO 111 K=1,M
  111 SOM = SOM + TAU(K,I)*S(K,J)
  113 SIM = SIM + SOM*XIXIO(J)
```


REPRODUCIBILITY
ORIGINAL PAGE IS POOR

```
      RH(N-I) = RH(N-I) + SIM
      RH(N-I) = AMIN1(RH(N-I),1.0)
  112  IF(RH(N-I).LE.0.) RH(N-I) = .01
C     WRITE(6,2010)(RH(I),I=1,NL)
C2010 FORMAT ('-RH IN ESTH2O'/(1X,10G13.5))
      SAM = 0.
      DO 114  I=1,NL
      QQ(I) = 622. * RH(I) * ES(I)/P(I)
      IF (P(I).LT.100.) QQ(I) = .001
114   SAM = SAM + QQ(I) * P(I)
C
C      THIS IS THE GM/CM**2 AT THE TOP OF THE ATMOSPHERE
C
      UTOTAL = DH*SAM/980.
C
C     WRITE(6,118) UTOTAL
C 118 FORMAT(1HO,'UTOTAL = ',F9.5)
C *** TEST FOR CONVERGENCE.
      IF(ABS(RMS-OLDRMS).LT.CVGH2O)RETURN
100   OLDRMS = RMS
      RETURN
      END
      SUBROUTINE ESTLSQ(XI,XLAM,M)
C *** GSFC PROGRAM NO. S00091   NIMBUS-4 TEMP./HUMIDITY ESTIMATION PROG
C
      COMMON/CHA/                    SUM,RMSH2O
      COMMON/ZZZ/XIO(22)   ,PHI(22,100)
C
      COMMON/XXX/  TS, PS, PT, DH, N, NL,                              R,
     #                                    MTOTAL,
     # P( 99),T1( 99),
     #                               HOTGRD
      LOGICAL                        HOTGRD
C
      DIMENSION XI(22),XLAM(22)
      DIMENSION H(22,22),XIXIO(22)
      REAL*8 H
      COMMON /H2O/ TO( 99),QQ( 99),TAU(28,100)
      DATA C1,C2 /3.7413E+04,1.4388E+04/
      DATA PI/3.14159/
C
      FSB(X,Y)=C1*(X**(-5))/(PI*(EXP(C2/(X*Y))-1.))
      DBDT(X,Y)=(FSB(X,Y)*C2*EXP(C2/(X*Y)))/(X*Y*Y*(EXP(C2/(X*Y))-1.))
C
      NLL = NL-1
```

```
      DO 10 I=1,NL
   10 T1(I)=T0(I)
      DO 100 L=1,3
      LL=L-1
C     WRITE(6,66) LL
C  66 FORMAT(1H0,53X,'TEMPERATURE PROFILE NUMBER ',I1)
      CALL RAD(T1,XLAM,T1(NL),M,N,XI0)
C     WRITE(6,150)
C 150 FORMAT(1H0,52X,'RESIDUALS (RADIANCES)')
C 151 FORMAT(60X,E13.4)
      SUM = 0.
      DO 101 I=1,M
      DEL = XI(I) - XI0(I)
      XIXI0(I) = DEL
C     WRITE(6,151) DEL
  101 SUM = SUM + (DEL*DEL)
      SUM = (SUM/M)**.5
C     WRITE(6,152) SUM
C 152 FORMAT(1H0,50X,'RMS RESIDUAL (RADIANCES) = ',E10.4)
      IF(L.EQ.3)  GO TO 1000
      DO 110 I=1,M
C
C       NLL IS THE NUMBER OF MIDPOINT LAYERS(?) - CHECK WITH DR. CURRAN
C
      DO 102 J=1,NLL
      PHI(I,J)=-DBDT(XLAM(I),T1(J) ) * (TAU(I,J+1) - TAU(I,J) )
  102 CONTINUE
      PHI(I,NL)=DBDT(XLAM(I),T1(NL)) * TAU(I,N-1)
  110 CONTINUE
      DO 104 I=1,M
      DO 104 K=1,M
      SOM = 0.
      DO 103 J=1,NL
C
C       THIS IS THE PRODUCT OF PHI AND PHI TRANSPOSE
C
  103 SOM = SOM + PHI(I,J)*PHI(K,J)
  104 H(I,K) = SOM
      DO 105 I=1,M
C
C       R IS THE DAMPING FACTOR,(TO BE SUPPLIED THRU BLOCK DATA),WHICH
C       I.REVAH SECURED FOR THE NIMBUS PROJECT.
  105 H(I,I) = H(I,I) + R
      CALL MATINT(H,M,0)
      DO 108 I=1,NL
```


REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

```
      SAM = 0.
      DO 109 J=1,M
      SIM = 0.
      DO 107 K=1,M
  107 SIM = SIM + PHI(K,I)*H(K,J)
  109 SAM = SAM + SIM * XIXIO(J)
  108 T1(I) = T1(I) + SAM
  100 CONTINUE
 1000 CONTINUE
      RETURN
      END
      SUBROUTINE GETSHL(IDT191,TS)
      COMMON /SHLDT  / TSVAL(100),TIMES(100)
C
C       THIS IS A DUMMY SUBROUTINE
C
      RETURN
      END
      SUBROUTINE H2ORAD(XLMH2O,DELH2O,XIH2O,XLMIN,XI,BEGWAV,DELWAV,
     #  NRADS,MH2O)
C *** GSFC PROGRAM NO. S00091   NIMBUS-4 TEMP./HUMIDITY ESTIMATION PROG
C *** COMPUTE RADIANCES FOR THE WATER VAPOR CALCS.
      DIMENSION XLMH2O(1),DELH2O(1),XIH2O(1),XLMIN(1),XI(1),
     # JX( 6),JY( 6),Z( 6)
      LOGICAL FSTIME/.TRUE./
C
C
      DO 100 I=1,MH2O
      IF(.NOT.FSTIME)GO TO 60
      JX(I) = ((XLMH2O(I)-DELH2O(I)/2.0) - BEGWAV) / DELWAV + 1.99999
      JY(I) = ((XLMH2O(I)+DELH2O(I)/2.0) - BEGWAV) / DELWAV + 0.99999
      Z(I) = JY(I) - JX(I) + 1
C
60    JSTART = JX(I)
      JSTOP = JY(I)
      X = 0
      DO 70 J=JSTART,JSTOP
70    X = X + XI(J)
      XIH2O(I) = X/Z(I)
C
100   CONTINUE
      FSTIME = .FALSE.
      RETURN
      END
      SUBROUTINE MATINH(A,N,M)
```

```
C ***.GSFC PROGRAM NO. S00091   NIMBUS-4 TEMP./HUMIDITY ESTIMATION PROG
C
C      MATRIX INVERSION WITH ACCOMPANYING SOLUTION OF LINEAR EQUATIONS
C
       IMPLICIT REAL*8(A-H,O-Z)
       DIMENSION IPIVOT( 9),A( 6, 6),B( 9,1),INDEX( 9,2),PIVOT( 9)
C
C   **** USED FOR HUMIDITY INVERSIONS ****
C      INITIALIZATION
C
    10 DETERM=1.0
    15 DO 20 J=1,N
    20 IPIVOT(J)=0
    30 DO 550 I=1,N
C
C      SEARCH FOR PIVOT ELEMENT
C
    40 AMAX=0.0
    45 DO 105 J=1,N
    50 IF (IPIVOT(J).EQ.1)GO TO 105
    60 DO 100 K=1,N
    70 IF (IPIVOT(K)-1) 80,100,740
    80 IF(DABS(AMAX).GE.DABS(A(J,K)))GO TO 100
    85 IROW=J
    90 ICOLUM=K
    95 AMAX=A(J,K)
   100 CONTINUE
   105 CONTINUE
   110 IPIVOT(ICOLUM)=IPIVOT(ICOLUM)+1
C
C      INTERCHANGE ROWS TO PUT PIVOT ELEMENT ON DIAGONAL
C
   130 IF(IROW.EQ.ICOLUM)GO TO 260
   140 DETERM=-DETERM
   150 DO 200 L=1,N
   160 SWAP=A(IROW,L)
   170 A(IROW,L)=A(ICOLUM,L)
   200 A(ICOLUM,L)=SWAP
   205 IF(M.LE.0)GO TO 260
   210 DO 250 L=1, M
   220 SWAP=B(IROW,L)
   230 B(IROW,L)=B(ICOLUM,L)
   250 B(ICOLUM,L)=SWAP
   260 INDEX(I,1)=IROW
   270 INDEX(I,2)=ICOLUM
```

```
  310 PIVOT(I)=A(ICOLUM,ICOLUM)
  320 DETERM=DETERM*PIVOT(I)
C
C     DIVIDE PIVOT ROW BY PIVOT ELEMENT
C
  330 A(ICOLUM,ICOLUM)=1.0
  340 DO 350 L=1,N
  350 A(ICOLUM,L)=A(ICOLUM,L)/PIVOT(I)
  355 IF(M.LE.0)GO TO 380
  360 DO 370 L=1,M
  370 B(ICOLUM,L)=B(ICOLUM,L)/PIVOT(I)
C
C     REDUCE NON-PIVOT ROWS
C
  380 DO 550 L1=1,N
  390 IF(L1.EQ.ICOLUM)GO TO 550
  400 T=A(L1,ICOLUM)
  420 A(L1,ICOLUM)=0.0
  430 DO 450 L=1,N
  450 A(L1,L)=A(L1,L)-A(ICOLUM,L)*T
  455 IF(M.LE.0)GO TO 550
  460 DO 500 L=1,M
  500 B(L1,L)=B(L1,L)-B(ICOLUM,L)*T
  550 CONTINUE
C
C     INTERCHANGE COLUMNS
C
  600 DO 710 I=1,N
  610 L=N+1-I
  620 IF (INDEX(L,1).EQ.INDEX(L,2))GO TO 710
  630 JROW=INDEX(L,1)
  640 JCOLUM=INDEX(L,2)
  650 DO 705 K=1,N
  660 SWAP=A(K,JROW)
  670 A(K,JROW)=A(K,JCOLUM)
  700 A(K,JCOLUM)=SWAP
  705 CONTINUE
  710 CONTINUE
  740 RETURN
      END
      SUBROUTINE MATINT(A,N,M)
C *** GSFC PROGRAM NO. S00091   NIMBUS-4 TEMP./HUMIDITY ESTIMATION PROG
C
C     MATRIX INVERSION WITH ACCOMPANYING SOLUTION OF LINEAR EQUATIONS
C
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION IPIVOT(33),A(22,22),B(33,1),INDEX(33,2),PIVOT(33)
C
C  **** USED FOR TEMPERATURE INVERSIONS ****
C     INITIALIZATION
C
   10 DETERM=1.0
   15 DO 20 J=1,N
   20 IPIVOT(J)=0
   30 DO 550 I=1,N
C
C     SEARCH FOR PIVOT ELEMENT
C
   40 AMAX=0.0
   45 DO 105 J=1,N
   50 IF (IPIVOT(J).EQ.1)GO TO 105
   60 DO 100 K=1,N
   70 IF (IPIVOT(K)-1) 80,100,740
   80 IF(DABS(AMAX).GE.DABS(A(J,K)))GO TO 100
   85 IROW=J
   90 ICOLUM=K
   95 AMAX=A(J,K)
  100 CONTINUE
  105 CONTINUE
  110 IPIVOT(ICOLUM)=IPIVOT(ICOLUM)+1
C
C     INTERCHANGE ROWS TO PUT PIVOT ELEMENT ON DIAGONAL
C
  130 IF(IROW.EQ.ICOLUM)GO TO 260
  140 DETERM=-DETERM
  150 DO 200 L=1,N
  160 SWAP=A(IROW,L)
  170 A(IROW,L)=A(ICOLUM,L)
  200 A(ICOLUM,L)=SWAP
  205 IF(M.LE.0)GO TO 260
  210 DO 250 L=1, M
  220 SWAP=B(IROW,L)
  230 B(IROW,L)=B(ICOLUM,L)
  250 B(ICOLUM,L)=SWAP
  260 INDEX(I,1)=IROW
  270 INDEX(I,2)=ICOLUM
  310 PIVOT(I)=A(ICOLUM,ICOLUM)
  320 DETERM=DETERM*PIVOT(I)
C
C     DIVIDE PIVOT ROW BY PIVOT ELEMENT
```

```
C
  330 A(ICOLUM,ICOLUM)=1.0
  340 DO 350 L=1,N
  350 A(ICOLUM,L)=A(ICOLUM,L)/PIVOT(I)
  355 IF(M.LE.0)GO TO 380
  360 DO 370 L=1,M
  370 B(ICOLUM,L)=B(ICOLUM,L)/PIVOT(I)
C
C     REDUCE NON-PIVOT ROWS
C
  380 DO 550 L1=1,N
  390 IF(L1.EQ.ICOLUM)GO TO 550
  400 T=A(L1,ICOLUM)
  420 A(L1,ICOLUM)=0.0
  430 DO 450 L=1,N
  450 A(L1,L)=A(L1,L)-A(ICOLUM,L)*T
  455 IF(M.LE.0)GO TO 550
  460 DO 500 L=1,M
  500 B(L1,L)=B(L1,L)-B(ICOLUM,L)*T
  550 CONTINUE
C
C     INTERCHANGE COLUMNS
C
  600 DO 710 I=1,N
  610 L=N+1-I
  620 IF (INDEX(L,1).EQ.INDEX(L,2))GO TO 710
  630 JROW=INDEX(L,1)
  640 JCOLUM=INDEX(L,2)
  650 DO 705 K=1,N
  660 SWAP=A(K,JROW)
  670 A(K,JROW)=A(K,JCOLUM)
  700 A(K,JCOLUM)=SWAP
  705 CONTINUE
  710 CONTINUE
  740 RETURN
      END
      SUBROUTINE NCLOUD(DAT191,IDT191)
      DIMENSION DAT191(448),XI(28)
      INTEGER*2 IDT191(896)
C
C
C *** GSFC PROGRAM NO. S00091   NIMBUS-4 TEMP./HUMIDITY ESTIMATION PROG
C
      COMMON/CHA/                    RMSTMP,RMSH2O
C
```

```
      COMMON/CO2/TAUCO2(28,100),CO2(28,9)   ,XK(28),XKMULT(28)
C
      COMMON /H2O/ TO( 99),OO( 99),TAU(28,100),WATER(28,9)
C
      COMMON/XXX/  TS, PS, PT, DH, N, NL,                              R,
     #                                    MTOTAL,
     # P( 99),T1( 99),
     #                          HOTGRD
      LOGICAL   SETP, ZEROVL,     HOTGRD
      DATA SETP / .FALSE. /
C
C
C       II IS USED IN THE CALL SEQUENCE TO ESTH2O. IT APPEARS
C       TO BE USED THERE IN THE CALL TO SUBROUTINE RADH2O. THERE
C       IT APPEARS TO BE USED IN THE CALCULATION OF THE PLANCK
C       RADIANCE FUNCTION,FSB, AS INDEX TO THE 'TO' ARRAY, INSTEAD
C       OF USING TS,(THE SHELTER OR SURFACE TEMPERATURE). HENCE,
C       THE VALUE OF  99.(THE LAST VALUE OR SURFACE VALUE).
C
      DATA II /  99 /
C
C       SIMILARLY, CVGH2O IS USED IN THE CALL TO ESTH2O. IT IS
C       THE TOLERANCE ALLOWED FOR CONVERGENCE,(I.E. IT IS THE
C       LIMIT ON THE DIFFERENCE OR CHANGE IN RMS VALUES CALCULATED
C       BEYOND WHICH CONVERGENCE IS NOT CONSIDERED TO HAVE TAKEN
C       PLACE.
C
      DATA CVGH2O / .001 /
C
C
C       RH2O IS A DAMPING FACTOR USED IN ESTH2O TO FACILITATE
C       CONVERGENCE OF MATRIX INVERSION. SEE PAPER BY DR. B.CONRATH.
C
C
      DATA RH2O/1.E-12/
C
      DIMENSION XLAM(28),XLMH2O( 6),DELH2O( 6),XIH2O( 6)
      DATA XLAM/ 15.,14.9,14.8,14.75,14.7,14.65,14.6,14.55,14.5,14.45,
     1  14.4,14.35,14.3,14.25,14.2,14.15,14.1,14.05,14.,13.95,
     2  13.9,13.8,      6.4,6.5,6.6,6.7,6.8,6.9  /
C
C       NOTE - THE FIRST TWELVE WAVELENGTHS ARE USED IN DETERMINING
C         TEMPERATURES,(IN TEMPERATURE INVERSION SECTION) , AND THE
C         LAST THREE ARE USED FOR HUMIDITY CALCULATIONS.
C
```


REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
C
      DATA XLMWIN/11.1/
C
C       XLMWIN IS THE WINDOW WAVELENGTH FOR DETERMINING IF HOT GROUND
C       CORRECTION IS NECESSARY
C
C
C*****
C       HUMIDITY ASSOCIATED WAVELENGTHS USED HERE WERE PREDICATED ON
C       THE JSC ORIGINAL PROMISE OF GOOD S-191 DATA IN THOSE AREAS.
C       SUCH DATA WAS NEVER PRESENT!
C*****
C
      EQUIVALENCE (XLMH20(1),XLAM(23))
      EQUIVALENCE (XI(23),XIH20(1))
      LOGICAL*1 LOND
C
      DATA MH20 /6/
      DATA C1,C2 /3.7413E+04,1.4388E+04/
      DATA PI/3.14159/
C
C
C       PLANCK RADIANCE FUNCTION.(NOTE-INVERSE PLANCK IS USED TO
C       CALCULATE EQUIVALENT BRIGHTNESS TEMPERATURES - THIS IS FORWARD
C       PLANCK)
C
      FSB(X,Y)=C1*(X**(-5))/(PI*(EXP(C2/(X*Y))-1.))
C
C       FTB IS THE INVERSE PLANCK FUNCTION REFERRED TO ABOVE.
C
      FTB(X,Y)=C2/(X*ALOG(1.+(C1/(PI*(X**5)*Y))))
C
C
C
C
C*****
C
C       NOTE THESE VALUES WILL BE CHANGED - FIRST, INDICES FOR
C                XI WILL BE ALTERED TO FIT DESIRES OF DR. CURRAN
C
C*****
C
C       THE FIRST ENTRY TO THIS PROGRAM CAUSES PRESSURE VALUES,
C       IN P, TO BE SET. SUBSEQUENT ENTRIES WILL BY-PASS THIS
C       LOOP.
```

```
C
      IF (SETP ) GO TO 10
      PSTEP=(PS-PT)/NL
      PRES=PT-PSTEP
      DO 5 L=1,NL
      PRES=PRES+PSTEP
      P(L)=PRES
    5 CONTINUE
      SETP=.TRUE.
   10 CONTINUE
C
      XI(1)=DAT191(444)
      XI(2)=DAT191(443)
      XI(3)=DAT191(442)
      XI(4)=DAT191(441)
      XI(5)=DAT191(440)
      XI(6)=DAT191(439)
      XI(7)=DAT191(438)
      XI(8)=DAT191(437)
      XI(9)=DAT191(436)
      XI(10)=DAT191(435)
      XI(11)=DAT191(434)
      XI(12)=DAT191(433)
      XI(13)=DAT191(432)
      XI(14)=DAT191(431)
      XI(15)=DAT191(430)
      XI(16)=DAT191(429)
      XI(17)=DAT191(428)
      XI(18)=DAT191(427)
      XI(19)=DAT191(426)
      XI(20)=DAT191(425)
      XI(21)=DAT191(424)
      XI(22)=DAT191(423)
      WINDOW=DAT191(396)
      XI(23)=DAT191(349)
      XI(24)=DAT191(350)
      XI(25)=DAT191(351)
      XI(26)=DAT191(352)
      XI(27)=DAT191(353)
      XI(28)=DAT191(354)
      ZEROVL=.FALSE.
      DO 15 L=23,28
      IF(XI(L).EQ.0.) ZEROVL=.TRUE.
   15 CONTINUE
C
```


REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
C
C      DEBUG
C
       WRITE(6,3000) XI
 3000 FORMAT(1HO,'XI VALUES INITIALLY WERE '/,1X,7E14.5/,1X,
     1 7E14.5/,1X,7E14.5/,1X,7E14.5/)
C
       WRITE(6,3100) R
 3100 FORMAT(1HO,'*****',' FOR THIS RUN THE VALUE OF R WAS ',
     1 E16.8,1X,'*****'/)
C
C
C       VALUES INITIALIZED THRU CALLS - WILL BE CHANGED
C
       CALL GETSHL(IDT191,TS)
C
       DH = (ALOG(PS)-ALOG(PT)) / (N-1.)
       CALL VIEWAN(DAT191,VWAN)
C
C       VIEW ANGLE WILL MODIFY INITIAL VALUES
C
       VN=VWAN*.0174533
       VWCOR=(1./COS(VN))
C
C
       CALL TRNSIN(PT,DH,MTOTAL,MH2O,N,MTOTAL)
       FSBWIN=FSB(XLMWIN,TS)
       DELTAI=WINDOW-FSBWIN
C
C       LIMITS BELOW WERE DERIVED BY DR.CURRAN - THE FIRST REJECTION
C       CRITERIUM IS BASED ON A 20. DEGREE TEMPERATURE DIFFERENCE.
C       THE SECOND HOTGROUND CORRECTION LIMIT IS BASED ON A 5 DEGREE
C       TEMPERATURE DIFFERENCE,(AS IS THE THIRD THOUGH NEGATIVE).
C       FURTHER CHANGES COULD CONCEIVABLY BE NEEDED. LATER ANALYSTS
C       MAY WISH TO CONSIDER SUCH LIMIT CHANGES.
C       DELTAI IS THE DIFFERENCE BETWEEN THE ACTUAL OR SENSED RADIANCE
C       AND THE PLANK CALCULATED VALUE,(BOTH AT THE 11.1 MICRON WINDOW)
C
       IF(DELTAI .LT. 2.23E-04 ) GO TO 20
       WRITE(6,2000)
 2000 FORMAT('+',T25,'REJECTING THIS SPECTRUM, DELTA I >=10.')
       RETURN
   20  CONTINUE
       IF(DELTAI .LT. 5.58E-05 ) GO TO 30
       HOTGRD= .TRUE.
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
      WRITE(6,2100)
 2100 FORMAT('+',T25,'NEED HOT GROUND CORRECTION.')
      GO TO 50
   30 CONTINUE
      IF(DELTAI.LE.-5.58E-05 ) GO TO 40
      HOTGRD=.FALSE.
      WRITE(6,2200)
 2200 FORMAT('+',T25,'DO NOT NEED HOT GROUND CORRECTION.')
      GO TO 50
   40 CONTINUE
      HOTGRD=.FALSE.
   50 CONTINUE
C
C*****
C
C       IN DEGREES KELVIN, THE EQUIVALENT TEMPERATURE IS CALCULATED
C        BY INVERSION OF THE MEASURED RADIANCES.
C
C*****
C
      TB= FTB(XLMWIN,WINDOW)
C
      XN = 0
      PC = 0
      OLDTOT = 99999.
      CALL TRANSW(TO,OO,MTOTAL,N,DH,PT,0,VWCOR)
      CALL TRANSC(TO,MTOTAL,N,DH,PT,0,VWCOR)
      DO 450 ITER=1,5
      IF(ITER .EQ. 1) GO TO 420
C *** RECOMPUTE TAU IF ESTH2O HAS USED IT.
      CALL TRANSW(T1,OO,MTOTAL,N,DH,PT,0,VWCOR)
      CALL TRANSC(T1,MTOTAL,N,DH,PT,0,VWCOR)
C
  420 CONTINUE
      DO 430 J=1,N
      DO 430 I=1,MTOTAL
  430 TAU(I,J) = TAU(I,J)*TAUCO2(I,J)
      IF(HOTGRD) GO TO 416
      DO 415 LB=1,MTOTAL
      XI(LB)=XI(LB)-(FSB(XLAM(LB),TB)-FSB(XLAM(LB),TS))*TAU(LB,N)
  415 CONTINUE
  416 CONTINUE
      WRITE(6,2400) HOTGRD
 2400 FORMAT(1HO,'XI VALUES AFTER CORRECTION LOOP - HOTGRD=',L4/)
      WRITE(6,2500) (I,XLAM(I),XI(I),I=1,MTOTAL)
```

```
 2500 FORMAT(1H ,'I=',I5,2X,'XLAM=',F11.5,2X, 'XI=',E15.8,5X,'I=',I5,2X,
     1  'XLAM=',F11.5,2X,'XI=',E15.8)
       CALL ESTLSO(XI,XLAM,MTOTAL)
C
  405 CONTINUE
C
C       IF ANY MIXING RATIO(HUMIDITY) ASSOCIATED RADIANCES ARE ZERO
C       SKIP THE CALCULATIONS,(MATRIX INVERSION WOULD BE IMPOSSIBLE!)
C
       IF(ZEROVL) GO TO 450
C *** COMPUTE WATER VAPOR.
       CALL ESTH2O (XIH2O,XLMH2O,MH2O,UTOTAL,T1,TB,II,RH2O,CVGH2O,
     1    VWCOR,MTOTAL)
C
C****
C       VERIFY THE VALUES BEING USED FOR II AND CVGH2O BEFORE USING THE
C       ROUTINE. SEE DRS. CONRATH OR CURRAN,OR H. POWELL(PMI).
C*****
C *** TEST FOR CONVERGENCE.
       IF(ABS(UTOTAL-OLDTOT)/UTOTAL .LE. CVGTH2)GO TO 460
       OLDTOT = UTOTAL
450    CONTINUE
       IF(.NOT. ZEROVL) GO TO 460
C
C       SINCE NO HUMIDITY CALCULATIONS WERE MADE, UTOTAL WAS NOT SET.
C       SET IT USING FIRST GUESS VALUES.
C
       SAM=0.
       DO 455 LT=1,NL
  455  SAM=SAM+QQ(LT)*P(LT)
       UTOTAL=(DH*SAM)/980.
  460  CONTINUE
C
C       SUBROUTINE WRNCLD WILL BOTH LIST CALCULATED VALUES ON PRINTER
C       AND PLACE THOSE TABLES ON MICROFILM USING THE WOLFF PLOT PACKAGE.
C
C
C       RECALL THAT UTOTAL IS THE TOTAL AMOUNT OF WATER VAPOR IN THE
C       ATMOSPHERE
C
C
C**********
C       NOTE - CORRECTION FACTOR OF VWCOR MAY NEED TO BE USED HERE
C       BEFORE VALUES ARE PLACED ON TAPE AND PRINTER. LATER ANALYSTS
C       SHOULD DETERMINE WHETHER THIS IS DESIRABLE.
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
      RETURN
      END
      SUBROUTINE TRANSC(T,M,N,DH,PT,KTYPE,VWCOR)
C *** GSFC PROGRAM NO. S00091   NIMBUS-4 TEMP./HUMIDITY ESTIMATION PROG
C
C       THIS SUBROUTINE CALCULATES TRANSMISSION COEFFICIENTS AS A
C       FUNCTION OF PRESSURE LEVEL.
C
      DIMENSION T(100),EXPP(28, 99)
C
      COMMON/C02/ TAU(28,100),C0(28),C1(28),C2(28),C3(28),C4(28),C5(28),
     # C6(28),C7(28),C8(28),XK(28),XKMULT(28)
C
      DO 11 K=1,M
       I=K+KTYPE
      TAVOLD = T(1)
      POLD = PT
      DO 11 J=2,N
      P = EXPP(I,J-1)
      TAV = (POLD*TAVOLD + (P - POLD)*T(J-1)) /P
      IF (C0(I) .NE. 0) GO TO 10
9     TAU(K,J) = 1.
      GO TO 1
10    U = 0.253*P
      PAV = .5 * P
      X = ALOG(U*273./TAV)
      Y = ALOG(PAV/1000.)
      Z = ALOG(TAV/273.)
C
C       NOTE - THIS IS A POLYNOMIAL FIT TO THE TRANSMISSION EQUATION
C       AS A FUNCTION OF ALTITUDE
C
       W=C0(I)+C1(I)*X + C2(I)*Y +C3(I)*Z +C4(I)*X*Y*.1 +C5(I)
     #   *(X*Z) + C6(I)*X**2*.1
     #   + C7(I)*X**2*Y*.01
     #   + C8(I)*(X*Z)*Z*.1
       W = XKMULT(I) * W
       IF(W.LT.-7.0) GO TO 3
       IF(W.GT.2.5)GO TO 4
        TAU(K,J) = EXP(-EXP(W)*VWCOR)
C
C       NOTE - EXP(W) IS THE ABSORPTION OPTICAL THICKNESS TO THE
C       PRESSURE LEVEL,PAV,(THE AVERAGE PRESSURE FOR THE CENTER
C       OF THE CURRENT LAYER). THE TRANSMISSION,TAU,IS EQUAL TO
C       THE OPTICAL THICKNESS TIMES THE SECANT OF THE VIEWING ANGLE.
```


PRECEDING PAGE BLANK NOT FILMED

```
C
      GO TO 1
    3 TAU(K,J) = 1.
      GO TO 1
    4 TAU(K,J) = 0.
1     POLD = P
      TAVOLD = TAV
11    CONTINUE
      DO 2 I=1,M
    2 TAU(I,1) = 1.
      RETURN
C
C*****************************    INITIALIZATION    **********************
      ENTRY TRNSIN(PT,DH,MTOTAL,MH2O,N,KTYPE)
C
      DO 900 J=2,N
      PP = ALOG(PT) + (J-1)*DH
      DO 800 I=1,MTOTAL
800   EXPP(I,J-1) = EXP(PP+XK(I))
      DO 900 I=1,MH2O
C
C       XK IS THE TRANSMISSION CORRECTION FACTOR ARRAY PASSED THRU
C       COMMON CO2
C
      EXPP(I+KTYPE,J-1)=EXP(PP+XK(I+KTYPE))
  900 CONTINUE
      RETURN
      END
      SUBROUTINE TRANSW(T,Q,M,N,DH,PT,KTYPE,VWCOR)
C *** GSFC PROGRAM NO. S00091   NIMBUS-4 TEMP./HUMIDITY ESTIMATION PROG
      COMMON /H2O/ TO( 99),QO( 99),TAU(28,100)
      COMMON/H2O/C0(28),C1(28),C2(28),C3(28),C4(28),C5(28),
     *  C6(28),C7(28),C8(28)
      DIMENSION T( 99),Q( 99),PP(100)
C *** "U" IS USED IN SUBROUTINE TQINIT.
      COMMON/UTOT/U
C
      LOGICAL FSTIME/.TRUE./
C
      IF(.NOT.FSTIME) GO TO 110
      FSTIME = .FALSE.
      APT = ALOG(PT)
      DO 100 J=1,N
      H=APT+(J-1)*DH
100   PP(J) = EXP(H)
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
C
110   CONTINUE
C
      PAVOLD = 0.
      TAVOLD = 0.
      POLD = PT
      UOLD = 0.
      DO 1 J=2,N
      P = PP(J)
      IF(O(J-1).EQ.0.) GO TO 2
      U = UOLD + O(J-1)*(P-POLD)/980.
      PAV = (UOLD*PAVOLD + .5*(P + POLD)*(U - UOLD))/U
      TAV = (UOLD*TAVOLD + T(J - 1)*(U - UOLD))/U
      GO TO 3
    2 CONTINUE
      DO 4 I=1,M
    4 TAU(I,J) = 1.
      POLD = P
      GO TO 1
    3 CONTINUE
      X = ALOG(U*273./TAV)
      Y = ALOG(PAV/1000.)
      Z = ALOG(TAV/273.)
      PAVOLD = PAV
      TAVOLD = TAV
      POLD = P
      UOLD = U
      DO 1 K=1,M
      I = K+KTYPE
      IF(C0(I).NE.0)GO TO 10
    9 TAU(K,J) =1.
      GO TO 1
10    W = C0(I) + C1(I)*X + C2(I)*Y + C3(I)*Z + C4(I)*X*Y*.1 + C5(I)*X
     # *Z + C6(I)*X*X*.1 +C7(I)*X*X*Z*.1 + C8(I)*Y*Z*Z*.1
      IF(W.LT.-7.0) GO TO 7
      IF(W.GT.2.5) GO TO 8
C
C       AS IN TRANSC, THE SECANT OF THE VIEW ANGLE MUST BE MULTIPLIED
C       BY THE ABSORPTION OPTICAL THICKNESS TO YIELD THE REQUIRED VALUE
C       FOR THE TRANSMISSION COEFFICIENT,TAU. HERE, WE CALCULATE VALUES F
C       THE MIXING RATIO,(HUMIDITY MEASURE), WHILE IN TRANSC, WE DO
C       SO FOR CARBON DIOXIDE,(TO DETERMINE TEMPERATURE VALUES).
C
      TAU(K,J) = EXP(-EXP(W)*VWCOR)
      GO TO 1
```

```
    7 TAU(K,J) =1.
      GO TO 1
    8 TAU(K,J) = 0.
    1 CONTINUE
      DO 6 I=1,M
    6 TAU(I,1) = 1.
      RETURN
      END
      SUBROUTINE VIEWAN(DAT191,VWAN)
      DIMENSION DAT191(448)
C     REAL*4 LEFT
C     EQUIVALENCE (DAT191(11),DOWN),(DAT191(12),LEFT),(DAT191(13),
C    1 UP),(DAT191(14),RIGHT)
C
C       EQUIVALENCING WAS PERFORMED HERE MERELY AS A DOCUMENTATION
C        OF THE ORDER IN WHICH THE POINTING ANGLES ARE RECORDED
C
      DIMENSION THETA(2)
      DATA CNDGRD / .0174533 /
      DATA CNRDDG / 57.29578 /
C
      LX=1
      THETA(1)=0.
      THETA(2)=0.
      DO 10 L=11,14
      IF(DAT191(L) .EQ. 0.) GO TO 10
      THETA(LX)=DAT191(L)
      LX=LX+1
      IF(LX.EQ.2) GO TO 20
   10 CONTINUE
C
   20 CONTINUE
      IF( (THETA(1) .EQ. 90. ) .OR. (THETA(2) .EQ. 90. ) ) GO TO 30
      ANG1=THETA(1)*CNDGRD
      ANG2=THETA(2)*CNDGRD
      TA1=TAN(ANG1)
      TA2=TAN(ANG2)
      TA1SQ=TA1*TA1
      TA2SQ=TA2*TA2
      SUM=TA1SQ + TA2SQ
      ARG=SQRT(SUM)
      VWANR=ATAN(ARG)
      VWAN=VWANR*CNRDDG
      RETURN
   30 CONTINUE
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
      VWAN=90.
      RETURN
      END
      SUBROUTINE WRNCLD(XLAM,XI,T1,RH2O,UTOTAL,TS,DAT191,IDT191,
     1 VIEWAN,ZEROVL)
C
C     THIS SUBROUTINE WAS WRITTEN BY M.MACK OF CSC FOR TASK 092
C     SKYLAB, UNDER CONTRACT NAS 5 - 11999 IN MAY, 1974. IT LISTS THE
C     CALCULATED TEMPERATURE AND HUMIDITY PROFILE VALUES CALCULATED
C     USING NCLOUD AND ITS ASSOCIATED RADIATIVE TRANSFER SUBROUTINES.
C     THESE LISTINGS APPEAR ON THE PRINTER AND THE ARRAYS ARE STORED ON
C     MAGNETIC TAPES FOR FURTHER SUBSEQUENT ANALYSIS PROGRAMMING.
C     IT ALSO UTILIZES THE WOLF PLOT PACKAGE TO DISPLAY VALUES ON THE
C     CALCOMP OR SC4020.
C
      DIMENSION XLAM(28),XI(28),T1( 99),RH2O( 99),DAT191(448),
     1 OUTAR(213),IOUTAR(213)
      INTEGER*2 IDT191(896)
      LOGICAL*1 PLOTON(100,43)
      EQUIVALENCE (OUTAR(1),IOUTAR(1)),(PLOTON(1,1),PLT1(1,1)),
     1 (PLOTON(1,23),PLT2(1,1))
      LOGICAL*1   PLT1(100,22)/128*' ','N','O',' ','C','L',
     1 'O','U','D',' ','S','P','E','C','T','R','U','M',' ','S',
     2 'E','N','S','E','D',  ' ',9*'0',' ',9*'0',232*' ','S',
     3 'H','E','L','T','E','R',' ','T','E','M','P','E','R','A',
     4 'T','U','R','E',' ','(','D','E','G','.',' ','A','B','S',
     5 '.',')',':',5*'0',5*' ','P','O','I','N','T','I','N','G',
     6 ' ','A','N','G','L','E','S',' ','(','D','E','G','.',')',
     7 ':',5*'0',',',5*'0',',',5*'0',' ','A','N','D',' ',5*'0',
     8 108*' ','N','A','D','I','R',' ','V','I','E','W',' ','A',
     9 'N','G','L','E',' ','(','D','E','G','.',')',':',5*'0',3*' ',
     A 'T','O','T','A','L',' ','W','A','T','E','R',' ','V','A',
     B 'P','O','R',' ','M','A','S','S',':',15*'0',3*' ','N','O',
     C '.',' ','O','F',' ','L','E','V','E','L','S',':',5*'0',245*' ',
     D 'T','E','M','P','E','R','A','T','U','R','E',' ','V','A','L',
     E 'U','E','S',241*' ',200*' ',200*' ',200*' ',200*' ',
     F 200*' '/
      LOGICAL*1 PLT2(100,21)/243*' ','H','U','M','I','D','I','T','Y',
     1 ' ','V','A','L','U','E','S',242*' ',200*' ',200*' ',200*' ',
     2 200*' ',200*' ',200*' ',200*' ',200*' '/
      LOGICAL ZEROVL
      LOGICAL*1 OKAY(100)/100*' '/
      LOGICAL*1 PROBLM(100,2) /39*' ','I','G','N','O','R','E',
     1 ' ','H','U','M','I','D','I','T','Y',' ','V','A','L','U','E',
     2 'S',':',38*' ',6*' ','T','E','M','P','E','R','A','T','U','R',
```

```
     2 'E','S',' ','D','E','R','I','V','E','D',' ','U','S','I','N',
     3 'G',' ','F','I','R','S','T',' ','G','U','E','S','S',' ','H',
     4 'U','M','I','D','I','T','Y',' ','D','I','S','T','R','I','B',
     5 'U','T','I','O','N',' ','B','E','C','A','U','S','E',' ','O',
     6 'F',' ','7','E','R','O',' ','R','A','D','I','A','N','C','E',
     7 'S','!',7*' '/
C
      DATA NLINES/43/
      DATA XO/.2/
      DATA YTOPSZ/ 8.0/
      DATA NLEVEL / 99 /
C
C
C       DEBUG INFORMATION
C
C
      DO 10 L=1,6
   10 IOUTAR(L)=IDT191(L)
      OUTAR(7)=TS
      OUTAR(8)=DAT191(11)
      OUTAR(9)=DAT191(12)
      OUTAR(10)=DAT191(13)
      OUTAR(11)=DAT191(14)
      OUTAR(12)=VIEWAN
      OUTAR(13)=UTOTAL
      IOUTAR(14)=NLEVEL
      I=0
      DO 20 LB=15,113
      I=I+1
      LB1=LB+ 99
      OUTAR(LB)=T1(I)
      OUTAR(LB1)=RH2O(I)
   20 CONTINUE
C
      IOUTAR(213)=1
      IF(ZEROVL) IOUTAR(213)=0
      DO 25 LD=1,100
      PLOTON(LD,39)=OKAY(LD)
      PLOTON(LD,41)=OKAY(LD)
      IF(ZEROVL) PLOTON(LD,39)=PROBLM(LD,1)
      IF(ZEROVL) PLOTON(LD,41)=PROBLM(LD,2)
   25 CONTINUE
C
C       NOTE - IOUTAR(213)=1 MEANS MIXING RATIOS(HUMIDITY VALUES)
C                            WERE PROPERLY CALCULATED
```


REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

```
C          IOUTAR(213)=0 MEANS HUMIDITY VALUES SHOULD BE IGNORED.
C                        THE TEMPERATURE FIELD WAS DERIVED USING
C                        FIRST GUESS HUMIDITY DISTRIBUTION
C                        BECAUSE OF ZERO RADIANCES.
C
C       FIRST, OUTPUT TAPE IS WRITTEN
C
      WRITE(10) OUTAR
C
C       NEXT, PRINTER OUTPUT IS CREATED
C
      WRITE(9,2000) (IOUTAR(I),I=1,6),(OUTAR(J),J=7,13),IOUTAR(14),
     1 (OUTAR(J),J=15,213),(PLOTON(I,39),I=1,100),(PLOTON(I,41),
     1 I=1,100)
 2000 FORMAT(1H1,45X,'NO CLOUD SPECTRUM SENSED',3I3,1X,3I3//,1X,2X,
     1 'SHELTER TEMPERATURE (DEG. ABS.):',F5.1,3X,'POINTING ANGLES',
     2 '(DEG.):',F5.1,',',F5.1,',',F5.1,' AND ',F5.1,2X,'NADIR VIEW',
     3 ' ANGLE(DEG.):',F5.1/,26X,'TOTAL WATER VAPOR MASS:',E15.7,25X,
     4 'NO. OF LEVELS:',I5//,58X,'TEMPERATURE VALUES'//, 9(1X,
     5 10(F8.3,5X)/),1X,9(F8.3,5X)/,//,1X,59X,'HUMIDITY VALUES'//,
     6  9(1X,10(F8.5,5X)/),1X,9(F8.5,5X)//,1X,62X,'ZEROVL=',I2/,10X,
     7  1X,100A1/,11X,100A1//)
C
C       NOW OUTPUT IS DISPLAYED BY WOLF ON CALCOMP OR SC4020
C
C
      CALL SAMSIZ(1)
C
      K=51
      DO 30 L=1,6
      K=K+3
      IF(L.EQ.4) K=K+1
      CALL EDIT(IOUTAR(L),'I3/',PLOTON(K,2),N,IBL)
   30 CONTINUE
      CALL EDIT(OUTAR(7),'F5.1/',PLOTON(37,5),N,IBL)
      CALL EDIT(OUTAR(8),'F5.1/',PLOTON(70,5),N,IBL)
      CALL EDIT(OUTAR(9),'F5.1/',PLOTON(76,5),N,IBL)
      CALL EDIT(OUTAR(10),'F5.1/',PLOTON(82,5),N,IBL)
      CALL EDIT(OUTAR(11),'F5.1/',PLOTON(92,5),N,IBL)
      CALL EDIT(OUTAR(12),'F5.1/',PLOTON( 29,7),N,IBL)
      CALL EDIT(OUTAR(13),'E15.7/',PLOTON(60,7),N,IBL)
      CALL EDIT(IOUTAR(14),'I5/',PLOTON( 92,7),N,IBL)
C
C       I WILL BE TEMPERATURES FIRST LINE
C
```

```
      J=11
      M=14
      DO 40 K=1,11
      I=I+1
      J=-8
      DO 40 L=1,9
      J=J+11
      M=M+1
      IF(M.EQ. 114) GO TO 40
      CALL EDIT(OUTAR(M),'F8.3/',PLOTON(J,I),N,IBL)
   40 CONTINUE
C
C       I HERE WILL BE HUMIDITY'S LINE
C
      I=26
      M=113
      DO 50 K=1,11
      I=I+1
      J=-8
      DO 50 L=1,9
      J=J+11
      M=M+1
      IF(M.EQ. 213) GO TO 50
      CALL EDIT(OUTAR(M),'F8.5/',PLOTON(J,I),N,IBL)
   50 CONTINUE
C
C
      STEP=(YTOPSZ -1.)/(NLINES+2)
      YO=YTOPSZ-2.5*STEP
      DO 60 NLINE=1,NLINES
      YO=YO-STEP
      CALL HORLIN(PLOTON(1,NLINE),100,XO,YO,100,0)
   60 CONTINUE
      CALL FRMADV
      RETURN
      END
      BLOCK DATA
C
C       NOTE - THE FOLLOWING ARE THE COMMONS AND THE SUBROUTINES USING
C       THEM
C
C       CHA     ESTH2O,ESTLSO,NCLOUD
C       CO2     BLDATTHP,ESTH2O,NCLOUD,TRANSC
C       H2O     BLDATTHP,DTAUDU,ESTH2O,ESTLSO,GETQO,GETTO,NCLOUD,
C               RAD,RADH2O,TRANSW
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
C      . SHLDT    BLDATTHP,GETSHL
C       UTOT      TRANSW
C       XXX       BLDATTHP,ESTH2O,ESTLSQ,NCLOUD
C       ZZZ       BLDATTHP,ESTLSQ
C
C       CORRECTIONS TO ANY OF THE COMMONS MUST BE MADE IN ALL
C       ROUTINES USING THEM
C
C ***************************************************************
C
      COMMON /H2O/ TO( 99),A1(60),A2(39),TAU(28,100),H2O1(60),H2O2(60),
     1 H2O3(60),H2O4(60),H2O5(12)
C
C       A1(60) AND A2(39) ARE EQUIVALENT,STORAGE-WISE, TO OO(99)
C       IT WAS EASIER TO SET UP THE VALUES THIS WAY DUE TO THE
C       OUTPUT SOURCE FROM WHICH IT WAS TAKEN.
C
C
C       NOTE: H2O1(60),.....,H2O4(60),H2O5(12),  STORAGE-WISE,
C             WILL BE THE SAME AS EITHER WATER(28,9) OR CO(28),
C             C1(28),C2(28),C3(28),C4(28),C5(28),C6(28),C7(28),AND
C             C8(28). AS INPUT FROM A DIFFERENT SOURCE,THIS DATA
C             SET-UP WAS THE QUICKEST.
C
      COMMON /SHLDT  / TSVAL(100),TIMES(100)
      COMMON/XXX/  TS, PS, PT, DH, N, NL,                          R,
     #                                    MTOTAL,
     # P( 99),T1( 99),
     #                                HOTGRD
C
C      XK VALUES,(USED AS TRANSMISSION CORRECTION FACTORS IN TRANSIN
C       ENTRY TO TRANSC),ARE NOT EXACT. CURRENT VALUES SET HERE
C       ARE TEMPORARY DATA,(UNALTERING ONES).                 THEY
C       ARE "FUDGE FACTORS" DETERMINED FOR NIMBUS IRIS DATA BY
C       I.REVAH BY COMPARING TEMPERATURES CALCULATED WITH THOSE
C       AVAILABLE THRU RADIOSOND DATA.
C
C
      COMMON/CO2/TAUC(28,100),CO21(60),CO22(60),CO23(60),CO24(60),
     # CO25(12),                        XK(28),XKMULT(28)
C
C       AS WITH COMMON H2O ARRAYS, USE OF CO21(60),CO22(60),
C       CO23(60),CO24(60,ANDCO25(12) INSTEAD OF EITHER
C       CARBON(28,9) OR CO(28),C1(28),C2(28),C3(28),C4(28),
C       C5(28),C6(28),C7(28) AND C8(28) MAKES NO DIFFERENCE
```

```
C       TO STORAGE ORDER.
C
        COMMON/ZZZ/XIO(22)   ,PHI(22,100)
C
C       NOTE THAT 'C' VALUES,(IN COMMON CO2 AND,CALLED H2O, IN COMMON
C       H2O), WILL BE SUPPLIED BY INTERPOLATION PROGRAM INTERPCV.
C
        LOGICAL HOTGRD
C
C       N IS THE NO. OF BOUNDARIES OR LEVELS UNDER CONSIDERATION
C
C       NL IS THE NUMBER OF LEVELS OR LAYERS.
C       MTOTAL REFERS TO THE NUMBER OF WAVELENGTHS ASSOCIATED WITH
C       TEMPERATURE CALCULATIONS.  MH2O,(NOT REFERENCED HERE BUT FOUND
C       IN SUBROUTINE NCLOUD),REFERS TO THOSE ASSOCIATED WITH HUMIDITY.
C       NOTE THAT HUMIDITY VALUES WILL ACTUALLY BE GIVEN AS MIXING RATIOS
C
        DATA MTOTAL,N/22,100/
        DATA  NL/99/
C
C       PT IS A PRESSURE AT THE TOP OF THE ATMOSPHERE
C       PS IS THE PRESSURE AT THE SURFACE
C
        DATA PS,PT/ 1013...05 /
C
C
        DATA XIO / 22*0. /
C
        DATA XK /28*0. /
        DATA XKMULT / 28*1. /
C
        DATA R /4.E-11/
C
C       R IS A DUMMY VALUE NOW. IT IS A DISPLACEMENT FOR
C       THE ARRAY H BEFORE IT IS INVERTED BY MATINV.
C       IT IS THE GAMMA VALUE IN DR. CONRATH'S PAPER.
C
C
C       VALUES SET FOR TSVAL AND TIMES ARRAYS ARE DUMMYS - SINCE HOUSTON
C       JSC VERIFIED THAT S-191 DATA WAS INVALID FOR INVERSION
C       PURPOSES, NO EXACT VALUES WERE SECURED.
        DATA TSVAL /100*0./
        DATA TIMES / 100*0./
C
C       THE FOLLOWING VALUES ARE ALSO NOT EXACT BUT ARE BEING
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
C     . USED TEMPORARILY FOR DEBUG PURPOSES - VALUES OF TO AND A1,A2,
C       (OO),AS WELL AS COEFFICIENTS ASSOCIATED WITH TEMPERATURE
C       ARE VALID,(ATSLEAST TO THE ACCURACY OF LINEAR INTERPOLATION
C       ANYWAY),BUT HUMIDITY ASSOCIATED VALUES ARE NOT.
C
      DATA TS / 294.  /
C
      DATA  TO  /
     X250.00,238.81,228.96,223.32,220.89,218.98,217.81,216.81,216.00,
     X216.00,216.00,216.00,216.00,216.00,216.00,216.00,216.00,216.00,
     X217.04,219.09,221.14,223.22,225.32,227.43,229.41,231.03,232.64,
     X234.26,235.90,237.57,239.23,240.90,242.43,243.71,244.99,246.27,
     X247.55,248.86,250.19,251.51,252.84,254.16,255.37,256.38,257.38,
     X258.39,259.40,260.40,261.37,262.29,263.21,264.12,265.04,265.96,
     X266.87,267.71,268.54,269.37,270.20,271.03,271.86,272.69,273.47,
     X274.22,274.97,275.72,276.46,277.21,277.96,278.71,279.41,280.08,
     X280.74,281.41,282.08,282.75,283.41,284.08,284.75,285.32,285.83,
     X286.34,286.85,287.36,287.88,288.39,288.90,289.41,289.92,290.31,
     X290.68,291.05,291.42,291.79,292.16,292.52,292.89,293.26,293.63/
      DATA     A1    /
     X 0.2065237E-02, 0.2270666E-01, 0.2137730E-01, 0.1314905E-01,
     X 0.8217081E-02, 0.6292060E-02, 0.4693240E-02, 0.4306145E-02,
     X 0.3822062E-02, 0.3686225E-02, 0.3608274E-02, 0.3565758E-02,
     X 0.3593048E-02, 0.3671609E-02, 0.3870139E-02, 0.4102785E-02,
     X 0.4963525E-02, 0.5824268E-02, 0.8353610E-02, 0.1248336E-01,
     X 0.1661311E-01, 0.2552864E-01, 0.3793259E-01, 0.5033654E-01,
     X 0.6605858E-01, 0.9145248E-01, 0.1168459E 00, 0.1422396E 00,
     X 0.1671709E 00, 0.1917105E 00, 0.2162501E 00, 0.2407897E 00,
     X 0.2674802E 00, 0.2983670E 00, 0.3292539E 00, 0.3601406E 00,
     X 0.3910275E 00, 0.4304689E 00, 0.4745664E 00, 0.5186639E 00,
     X 0.5627614E 00, 0.6068589E 00, 0.6536110E 00, 0.7048858E 00,
     X 0.7561607E 00, 0.8074355E 00, 0.8587103E 00, 0.9099852E 00,
     X 0.9681547E 00, 0.1036329E 01, 0.1104504E 01, 0.1172679E 01,
     X 0.1240853E 01, 0.1309029E 01, 0.1377203E 01, 0.1504305E 01,
     X 0.1641026E 01, 0.1777746E 01, 0.1914467E 01, 0.2051188E 01/
      DATA     A2    /
     X 0.2187908E 01, 0.2324629E 01, 0.2481635E 01, 0.2650696E 01,
     X 0.2819756E 01, 0.2988816E 01, 0.3157877E 01, 0.3326938E 01,
     X 0.3495998E 01, 0.3665058E 01, 0.3888401E 01, 0.4146021E 01,
     X 0.4403642E 01, 0.4661263E 01, 0.4918883E 01, 0.5176503E 01,
     X 0.5434124E 01, 0.5691744E 01, 0.5949365E 01, 0.6209929E 01,
     X 0.6472287E 01, 0.6734645E 01, 0.6997004E 01, 0.7259361E 01,
     X 0.7521719E 01, 0.7784076E 01, 0.8046433E 01, 0.8308792E 01,
     X 0.8571150E 01, 0.8856753E 01, 0.9146533E 01, 0.9436313E 01,
     X 0.9726092E 01, 0.1001587E 02, 0.1030565E 02, 0.1059543E 02,
```

```
     X 0.1088521E 02, 0.1117499E 02, 0.1146477E 02/
C
C       THE TO AND OO(A1 AND A2) VALUES,(TEMPERATURE AND MIXING
C       RATIO VALUES), ARE FOR MIDLATITUDE SUMMER. REPLACE WITH
C       MIDLATITUDE WINTER,(STORED IN ZBMJM.DATATOIN - 2ND TWO
C       BLOCK DATA PROGRAMS - I.E. 3RD AND 4TH).
C
C       REMEMBER AGAIN!!!!!
C       TRANSMISSION COEFFICIENTS IN H2O1,ETC.,AND CO21,ETC. ARE
C       NOT ACCURATE FOR HUMIDITY(MIXING RATIO) RELATED WAVELENGTHS.
C
C
      DATA H2O1 /
     X  0.6616435E 00,  0.6661456E 00,  0.6844586E 00,  0.6997145E 00,
     X  0.7263946E 00,  0.7533212E 00,  0.7861198E 00,  0.8189184E 00,
     X  0.8466128E 00,  0.8737973E 00,  0.8498231E 00,  0.8137409E 00,
     X  0.7875854E 00,  0.7649484E 00,  0.8023340E 00,  0.8657542E 00,
     X  0.8707373E 00,  0.8489100E 00,  0.8537720E 00,  0.8699778E 00,
     X  0.8871260E 00,  0.9265140E 00,  0.3404921E 02,  0.3351137E 02,
     X  0.3297353E 02,  0.3243568E 02,  0.3189786E 02,  0.3136002E 02,
     X  0.7675267E 00,  0.7809180E 00,  0.8019611E 00,  0.8291678E 00,
     X  0.9388939E 00,  0.1037856E 01,  0.8803857E 00,  0.7229148E 00,
     X  0.7033198E 00,  0.6975116E 00,  0.6965562E 00,  0.6967494E 00,
     X  0.7595335E 00,  0.8444962E 00,  0.7866891E 00,  0.6669562E 00,
     X  0.6637520E 00,  0.7140102E 00,  0.8211963E 00,  0.9525806E 00,
     X  0.9836807E 00,  0.8881040E 00, -0.1653192E 02, -0.1625194E 02,
     X -0.1597196E 02, -0.1569199E 02, -0.1541201E 02, -0.1513203E 02,
     X  0.0          , -0.1514757E 00, -0.7538998E-01, -0.1791380E-01/
      DATA H2O2 /
     X -0.1061847E 00, -0.1872916E 00, -0.9771723E-01, -0.8142769E-02,
     X  0.0          ,  0.0          ,  0.0          ,  0.0          ,
     X  0.0          ,  0.0          ,  0.0          ,  0.0          ,
     X  0.0          ,  0.0          , -0.2536295E-01, -0.6150642E-01,
     X -0.4337224E-01,  0.0          , -0.2232638E 00, -0.2130252E 00,
     X -0.2027867E 00, -0.1925482E 00, -0.1823096E 00, -0.1720710E 00,
     X  0.4283252E 00,  0.4306476E 00,  0.4104365E 00,  0.4062430E 00,
     X  0.4517888E 00,  0.4953449E 00,  0.4914973E 00,  0.4876497E 00,
     X  0.4825824E 00,  0.4773931E 00,  0.4694475E 00,  0.4608496E 00,
     X  0.4508216E 00,  0.4402870E 00,  0.4177909E 00,  0.3901067E 00,
     X  0.3626016E 00,  0.3351789E 00,  0.3123992E 00,  0.2915924E 00,
     X  0.2839807E 00,  0.2909876E 00,  0.0          ,  0.0          ,
     X  0.0          ,  0.0          ,  0.0          ,  0.0          ,
     X -0.7394451E-01,  0.0          ,  0.0          ,  0.5516779E-01,
     X  0.3270091E 00,  0.5911075E 00,  0.6707345E 00,  0.7503615E 00/
      DATA H2O3 /
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
     X  0.71856l5E 00,  0.67561l5E 00,  0.6601741E 00,  0.6512381E 00,
     X  0.6748399E 00,  0.7099715E 00,  0.8073348E 00,  0.9316900E 00,
     X  0.8438424E 00,  0.6586375E 00,  0.6181083E 00,  0.6390684E 00,
     X  0.6395296E 00,  0.6863538E 00,  0.7301086E 01,  0.7195162E 01,
     X  0.7089238E 01,  0.6983314E 01,  0.6877390E 01,  0.6771465E 01,
     X  0.9332305E 00,  0.7982852E 00,  0.6249945E 00,  0.5134309E 00,
     X  0.2514386E 00,  0.0           ,  0.0           ,  0.0           ,
     X  0.1309086E 01,  0.2749104E 01,  0.3067351E 01,  0.3120098E 01,
     X  0.2000103E 01,  0.4645510E 00,  0.1246492E 01,  0.3033638E 01,
     X  0.3575034E 01,  0.3544889E 01,  0.2427858E 01,  0.8488445E 00,
     X  0.9508496E 00,  0.3083271E 01,  0.1128486E 02,  0.1114721E 02,
     X  0.1100956E 02,  0.1087191E 02,  0.1073426E 02,  0.1059661E 02,
     X  0.1554161E 01,  0.1128922E 01,  0.7762372E 00,  0.5894414E 00,
     X  0.3334140E 00,  0.8213717E-01, -0.5595475E-01, -0.1940468E 00,
     X -0.2738136E 00, -0.3477498E 00, -0.2089465E 00, -0.1979214E-01/
      DATA H204 /
     X  0.6425041E-01,  0.1110436E 00,  0.2868932E-01, -0.1096819E 00,
     X -0.6750190E-01,  0.5751383E-01,  0.2586727E 00,  0.4921994E 00,
     X  0.7342357E 00,  0.1256866E 01,  0.9335258E 02,  0.9198074E 02,
     X  0.9060892E 02,  0.8923708E 02,  0.8786525E 02,  0.8649341E 02,
     X  0.2922273E 01,  0.4161613E 01,  0.5674290E 01,  0.6494922E 01,
     X  0.8501526E 01,  0.1038631E 02,  0.9368804E 01,  0.8351295E 01,
     X  0.4684400E 01,  0.7524986E 00, -0.3148904E 00, -0.7043089E 00,
     X  0.1782732E 01,  0.5289037E 01,  0.4463647E 01,  0.1759394E 01,
     X  0.1051890E 01,  0.1260486E 01,  0.1189320E 01,  0.9992474E 00,
     X  0.1380103E 01,  0.1721099E 01, -0.1611418E 03, -0.1584989E 03,
     X -0.1558560E 03, -0.1532131E 03, -0.1505701E 03, -0.1479272E 03,
     X  0.2965899E 00,  0.4453796E 00,  0.9757605E 00,  0.1284554E 01,
     X  0.1779546E 01,  0.2230705E 01,  0.1637563E 01,  0.1044420E 01,
     X  0.5618125E 00,  0.9024912E-01,  0.6684099E 00,  0.1495018E 01,
     X  0.1914985E 01, -0.2190844E 01,  0.1425813E 01,  0.2093058E 00/
      DATA H205 /
     X -0.1601657E 00, -0.1410201E 00, -0.1352202E 00, -0.1350924E 00,
     X -0.8285499E-01,  0.0           , -0.4846269E 00, -0.4758294E 00,
     X -0.4670318E 00, -0.4582343E 00, -0.4494368E 00, -0.4406393E 00/
      DATA C021 /
     X  0.1179782E 01,  0.6386714E 00,  0.1784992E 00,  0.6429839E-01,
     X  0.3965280E-01,  0.1127647E-01, -0.1059820E 00, -0.2232405E 00,
     X -0.3607101E 00, -0.5002034E 00, -0.6853607E 00, -0.8813258E 00,
     X -0.1105533E 01, -0.1339744E 01, -0.1601852E 01, -0.1876060E 01,
     X -0.2164001E 01, -0.2458241E 01, -0.2237123E 01, -0.1796960E 01,
     X -0.1585274E 01, -0.2232044E 01, -0.9756708E 02, -0.9605638E 02,
     X -0.9454568E 02, -0.9303497E 02, -0.9152429E 02, -0.9001358E 02,
     X -0.4216824E 00, -0.1366339E 00,  0.8517232E 00,  0.1211899E 01,
     X  0.5934942E 00,  0.1455241E-01,  0.3758065E 00,  0.7370605E 00,
```

```
     X  0.9299919E 00,  0.1106095E 01,  0.1184560E 01,  0.1239915E 01,
     X  0.9852315E 00,  0.6206858E 00,  0.5937375E 00,  0.7132218E 00,
     X  0.8802454E 00,  0.1069079E 01,  0.1041454E 01,  0.9218285E 00,
     X  0.9285456E 00,  0.1022309E 01,  0.4304224E 00,  0.4320166E 00,
     X  0.4336107E 00,  0.4352049E 00,  0.4367990E 00,  0.4383932E 00,
     X  0.7857134E 00,  0.8032232E 00,  0.9580245E 00,  0.1012708E 01/
      DATA C022 /
     X  0.8575283E 00,  0.7083694E 00,  0.7026448E 00,  0.6969203E 00,
     X  0.6965731E 00,  0.6967635E 00,  0.7645276E 00,  0.8482848E 00,
     X  0.7830460E 00,  0.6650096E 00,  0.6638622E 00,  0.7134154E 00,
     X  0.8178081E 00,  0.9473607E 00,  0.9838475E 00,  0.9807801E 00,
     X  0.8972437E 00,  0.7749994E 00,  0.1608484E 02,  0.1580906E 02,
     X  0.1553328E 02,  0.1525750E 02,  0.1498172E 02,  0.1470593E 02,
     X  0.4615484E 00,  0.4429125E 00,  0.4016274E 00,  0.3762951E 00,
     X  0.3511409E 00,  0.3262306E 00,  0.3071333E 00,  0.2880359E 00,
     X  0.2837031E 00,  0.2808466E 00,  0.2900255E 00,  0.3020530E 00,
     X  0.2937676E 00,  0.2782841E 00,  0.2823358E 00,  0.2948608E 00,
     X  0.2852887E 00,  0.2655785E 00,  0.2473592E 00,  0.2297732E 00,
     X  0.2088337E 00,  0.1549294E 00,  0.6275913E 01,  0.6180664E 01,
     X  0.6085416E 01,  0.5990167E 01,  0.5894918E 01,  0.5799668E 01,
     X  0.1534361E 01,  0.4016352E 01,  0.1763455E 01,  0.7518265E 00,
     X  0.4456480E 01,  0.8009702E 01,  0.7955131E 01,  0.7900559E 01/
      DATA C023 /
     X  0.7370008E 01,  0.6791849E 01,  0.6798726E 01,  0.6944070E 01,
     X  0.6710688E 01,  0.6343103E 01,  0.5855285E 01,  0.5315319E 01,
     X  0.4825617E 01,  0.4358975E 01,  0.4259225E 01,  0.4315408E 01,
     X  0.3836330E 01,  0.2832808E 01,  0.4358382E 02,  0.4286339E 02,
     X  0.4214299E 02,  0.4142258E 02,  0.4070216E 02,  0.3998175E 02,
     X  0.2282374E 01,  0.2334950E 01,  0.1731441E 01,  0.1430619E 01,
     X  0.1669215E 01,  0.1875007E 01,  0.1299272E 01,  0.7235374E 00,
     X  0.8437567E 00,  0.1033571E 01,  0.9638166E 00,  0.8326260E 00,
     X  0.9117555E 00,  0.1065413E 01,  0.7264460E 00,  0.1738061E 00,
     X  0.1106887E 01,  0.2721606E 01,  0.2453109E 01,  0.1384193E 01,
     X  0.1471658E 01,  0.1934051E 01,  0.0          ,  0.0          ,
     X  0.0          ,  0.0          ,  0.0          ,  0.0          ,
     X  0.6375951E-01,  0.5541275E-01,  0.6485784E-01,  0.7120317E-01,
     X  0.7013506E-01,  0.6906646E-01,  0.6798726E-01,  0.6690806E-01,
     X  0.6486237E-01,  0.6272000E-01,  0.7237148E-01,  0.8481431E-01/
      DATA C024 /
     X  0.7999849E-01,  0.6906700E-01,  0.5513478E-01,  0.3990104E-01,
     X  0.3794010E-01,  0.4206865E-01,  0.4414662E-01,  0.4535311E-01,
     X  0.3270733E-01,  0.6934974E-02,  0.1223513E 00,  0.1217667E 00,
     X  0.1211823E 00,  0.1205977E 00,  0.1200132E 00,  0.1194287E 00,
     X  0.3736235E 00,  0.1210618E 00,  0.0          , -0.3331275E-02,
     X -0.1974626E-01, -0.3592340E-01, -0.4643397E-01, -0.5694454E-01,
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
     X -0.1726782E 00, -0.2989362E 00, -0.2781826E 00, -0.2226342E 00,
     X -0.1854522E 00, -0.1547794E 00, -0.1586384E 00, -0.1774756E 00,
     X -0.2250395E 00, -0.2857831E 00, -0.2978372E 00, -0.2891975E 00,
     X -0.2534490E 00, -0.1757344E 00, -0.3755959E 01, -0.3696843E 01,
     X -0.3637728E 01, -0.3578612E 01, -0.3519497E 01, -0.3460382E 01,
     X -0.2990954E 00, -0.5016104E 00, -0.4808236E 00, -0.3901930E 00,
     X -0.1910863E 00, -0.1021069E-01, -0.2636845E 00, -0.5171582E 00,
     X -0.3975928E 00, -0.2407224E 00, -0.4422004E 00, -0.7284920E 00,
     X -0.7400153E 00, -0.6541706E 00, -0.4425278E 00, -0.1763222E 00/
      DATA C025 /
     X -0.1052128E 00, -0.1236123E 00, -0.3262996E 00, -0.6073189E 00,
     X -0.6019962E 00, -0.5017326E 00, -0.1765769E 01, -0.1735494E 01,
     X -0.1705218E 01, -0.1674943E 01, -0.1644669E 01, -0.1614393E 01/
      END
// EXEC LINKGO,REGION.GO=350K
//*    UNIT 6 IS THE DEBUG AND GENERAL MESSAGE UNIT ON PRINTER
//*     UNIT 9 CONTAINS THE LISTING OF THE NO-CLOUD QUALIFYING DATA
//*      UNIT 10 CONTAINS THE LISTING OF THE NON-CLOUD,(CLEAR),
//*        QUALIFYING DATA-IT IS A TAPE!
//*       UNIT 12 IS THE INPUT DATA TAPE,(GSFC FORM OF S-191 DATA)
//LINK.SYSLIB DD DSN=SYS2.WOLFPLOT,DISP=SHR
//GO.FT06F001 DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=7265),
// SPACE=(CYL,(10,2))
//GO.FT09F001 DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=7265),
// SPACE=(CYL,(4,1))
//GO.FT10F001 DD DUMMY
//GO.FT12F001 DD UNIT=(2400-9,,DEFER),DCB=(RECFM=VBS,LRECL=1796,
// BLKSIZE=30536,DEN=3),LABEL=(,BLP),VOL=SER=DUMMY
//GO.PLOTTAPE DD DUMMY
//GO.SYSUDUMP DD SYSOUT=A
//GO.DATA5 DD *
Y-292
 &INPUT
 PLOTER='PRINTER ',MONTH=9,IDAY=11,IYEAR=73,IHOUR=13,MINUTE=10,
 ISEC=40,IENDHR=13,IENDMI=10,IENDSC=41,
 &END
// EXEC NOTIFY,MODE=ALL,MSG='JOB IS COMPLETE'
```

APPENDIX G - BLOCK DATA ROUTINE PROGRAM LISTINGS, INPUT, AND JCL OUTPUT

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
      SUBROUTINE BLKDAT(NTYPE,ARR4,ARR8,INTAR2,INTAR4,NOELEM,
     1   ARNAME,FMT,IFMTSZ)
C
C       NTYPE - 1=REAL*4  DATA IS IN ARR4
C               2=REAL*8  DATA IS IN ARR8
C               3=INTEGER*2 DATA IS IN INTAR2
C               4=INTEGER*4 DATA IS IN INTAR4
C       NOELEM IS THE NUMBER OF ELEMENTS IN THE ARRAY
C       ARNAME,(WHICH IS REAL*8), IS THE EBCDIC NAME OF THE ARRAY,
C           (FORTRAN CONVENTIONS:ONE TO SIX LETTERS BUT LEFT ADJUSTED).
C       FMT IS THE DESIRED DATA FORMAT: IN
C                                       EN.I
C                                       DN.I
C                                       FN.I
C                                       AN
C       IFMTSZ HAS THE VALUE N IN THE ABOVE EXAMPLES
C              (I.E. IT IS THE WIDTH OF THE DESIRED FORMAT)
C
      LOGICAL*1 FM1( 36),FM2( 32),FM3(140),FM4(136),FT1( 80),
     1 FT2( 80),FT3( 80),FT4( 80)
      REAL*4 ARR4(1),FORMS1( 8),FORMS2( 7),FORMS3(35),
     1 FORMS4(34),ARNMSR(100),ARNMSI(100),TYPES(12),FOUT1(20),
     2 FOUT2(20),FOUT3(20),FOUT4(20),FMT1(2)
      REAL*8 ARR8(1),FMT,ARNAME,FMTU
      INTEGER*2 INTAR2(1)
      INTEGER*4 INTAR4(1),INDEX(100)
      DATA FORMS1 / '(1H ',',6X,',',3A4,',',(1X,',','A8,''',',('',I',
     1 '6,'')',',',''))'                                    /
      DATA FORMS2 /         '(6X,',',3A4,',',(1X,',','A8,''',',('',I',
     1 '6,'')',',',''))'                                   /
      DATA FORMS3 / '(1H ',',5X,',''',X'' ',',    ',',    ',',A8,',
     1 '''('',',',I6,''',','),'',',','A8,''',',('',I',',6,'')',',',''/,',
     2 '(6X,',',''',X'',',','A8,''',',('',I',',6,'')',',','',A',
     3 '8,''('',''',I6',',','''),',',''',A8',',',''('''',',I6,',''''),''',
     4 ',A8,',''',('',',','I6,''',','),'',',','A8,''',',('',I',',6,'')',
     5 ',''/)';')    '  /
      DATA FORMS4 /         '(5X,',',''',X'' ',',    ',',    ',',A8,',
     1 '''('',',',I6,''',','),'',',','A8,''',',('',I',',6,'')',',',''/,',
     2 '(6X,',',''',X'',',','A8,''',',('',I',',6,'')',',','',A',
     3 '8,''('',''',I6',',','''),',',''',A8',',',''('''',',I6,',''''),''',
     4 ',A8,',''',('',',','I6,''',','),'',',','A8,''',',('',I',',6,'')',
     5 ',''/)',')    '  /
```

```
      DATA FOUT1 / '( 14','(6X,',''''X'',',','NN( ',',     ',',     ',
     1 ',','','')/),' ,'     ','6X, ',''''X'',',','NI( ',
     2 '     ',',     ',',','','),  ',',     ',',     ',',','''/''',
     3 '    )'  /
      DATA FOUT2 / '( 14','(5X,',''''X'',',','NN( ',',     ',',     ',
     1 ',','','')/),' ,'     ','5X, ',''''X'',',','NI( ',
     2 '     ',',     ',',','','),  ',',     ',',     ',',','''/''',
     3 '    )'  /
      DATA FOUT3 / '( KK','(6X,',''''X'',',','NN( ',',     ',',     ',
     1 ',','','')/),',','6X,',' ''X''',' ,  ','NI( ',',     ',',     ',
     2 ',','',''),  ',',     ',',     ',',''/''',',    )'  /
      DATA FOUT4 / '( KK','(5X,',''''X'',',','NN( ',',     ',',     ',
     1 ',','','')/),',','5X,',' ''X''',' ,  ','NI( ',',     ',',     ',
     2 ',','',''),  ',',     ',',     ',',''/''',',    )'  /
C
C
      EQUIVALENCE (FORMS1(1),FM1(1)),(FORMS2(1),FM2(1)),
     1 (FORMS3(1),FM3(1)),(FORMS4(1),FM4(1)),
     2 (FOUT1(1),FT1(1)),(FOUT2(1),FT2(1)),(FOUT3(1),FT3(1)),
     3 (FOUT4(1),FT4(1)),(FMT1(1),FMTU)
C
      DATA   TYPES   /'REAL','*4   ',',     ','REAL','*8   ',',     ',
     1 'INTE','GER*','2    ','INTE','GER*','4    ' /
      DATA ARNMSR / 'A1   ','A2   ','A3   ','A4   ','A5   ','A6   ','A7   ',
     1 'A8   ','A9   ','A10  ','A11  ','A12  ','A13  ','A14  ','A15  ','A16  ',
     2 'A17  ','A18  ','A19  ','A20  ','A21  ','A22  ','A23  ','A24  ','A25  ',
     3 'A26  ','A27  ','A28  ','A29  ','A30  ','A31  ','A32  ','A33  ','A34  ',
     4 'A35  ','A36  ','A37  ','A38  ','A39  ','A40  ','A41  ','A42  ','A43  ',
     5 'A44  ','A45  ','A46  ','A47  ','A48  ','A49  ','A50  ','A51  ','A52  ',
     6 'A53  ','A54  ','A55  ','A56  ','A57  ','A58  ','A59  ','A60  ','A61  ',
     7 'A62  ','A63  ','A64  ','A65  ','A66  ','A67  ','A68  ','A69  ','A70  ',
     8 'A71  ','A72  ','A73  ','A74  ','A75  ','A76  ','A77  ','A78  ','A79  ',
     9 'A80  ','A81  ','A82  ','A83  ','A84  ','A85  ','A86  ','A87  ','A88  ',
     A 'A89  ','A90  ','A91  ','A92  ','A93  ','A94  ','A95  ','A96  ','A97  ',
     8 'A98  ','A99  ','A100'  /
      DATA ARNMSI / 'I1   ','I2   ','I3   ','I4   ','I5   ','I6   ','I7   ',
     1 'I8   ','I9   ','I10  ','I11  ','I12  ','I13  ','I14  ','I15  ','I16  ',
     2 'I17  ','I18  ','I19  ','I20  ','I21  ','I22  ','I23  ','I24  ','I25  ',
     3 'I26  ','I27  ','I28  ','I29  ','I30  ','I31  ','I32  ','I33  ','I34  ',
     4 'I35  ','I36  ','I37  ','I38  ','I39  ','I40  ','I41  ','I42  ','I43  ',
     5 'I44  ','I45  ','I46  ','I47  ','I48  ','I49  ','I50  ','I51  ','I52  ',
     6 'I53  ','I54  ','I55  ','I56  ','I57  ','I58  ','I59  ','I60  ','I61  ',
     7 'I62  ','I63  ','I64  ','I65  ','I66  ','I67  ','I68  ','I69  ','I70  ',
     8 'I71  ','I72  ','I73  ','I74  ','I75  ','I76  ','I77  ','I78  ','I79  ',
     9 'I80  ','I81  ','I82  ','I83  ','I84  ','I85  ','I86  ','I87  ','I88  ',
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
     A 'I89 ','I90 ','I91 ','I92 ','I93 ','I94 ','I95 ','I96 ','I97 ',
     B 'I98 ','I99 ','I100'  /
C
C       FORMATS
C
 2000 FORMAT(1H1,6X,'BLOCK DATA')
 2050 FORMAT(6X,'BLOCK DATA')
 2100 FORMAT(1H ,6X,3A4,1X,A8,'(',I6,')')
 2150 FORMAT(6X,3A4,1X,A8,'(',I6,')')
 2200 FORMAT(1H ,5X,'X',1X,A8,'(',I6,')')
 2250 FORMAT(5X,'X',1X,A8,'(',I6,')')
 2300 FORMAT(1H ,6X,'EQUIVALENCE ')
 2350 FORMAT(     6X,'EQUIVALENCE ')
 2400 FORMAT(6X,'X','(',A8,'(',I6,'),',A8,'(1)),',(',A8,'(',I6,
    1 '),',A8,'(1)),')
 2450 FORMAT(5X,'X','(',A8,'(',I6,'),',A8,'(1)),',(',A8,'(',I6,
    1 '),',A8,'(1)),')
 2500 FORMAT(6X,'X','(',A8,'(',I6,'),',A8,'(1)),',
    1 '(',A8,'(',I6,'),',A8,'(1))')
 2550 FORMAT(5X,'X','(',A8,'(',I6,'),',A8,'(1)),',
    1 '(',A8,'(',I6,'),',A8,'(1))')
 2600 FORMAT(1H ,5X,'X',1X,'(',A8,'(',I6,'),',A8,'(1))')
 2650 FORMAT(     5X,'X',1X,'(',A8,'(',I6,'),',A8,'(1))')
 2700 FORMAT(1H ,6X,'DATA ',A8,2X,'/')
 2750 FORMAT(6X,'DATA ',A8,2X,'/')
 2800 FORMAT(1H ,6X,'END')
 2850 FORMAT(6X,'END')
C
C       SET INPUT FORMAT INTO WORK AREA EQUIVALENCED FORMAT SPACE
C
      FMTU=FMT
C
C       CALCULATE NO. ELEMENTS ON LINE, NOONLN; NO. OF LINES,NOLINS:
C       NO. OF ARRAYS WHICH ARE THE SAME LENGTH,NOARSM:TOTAL NO. OF
C       ARRAYS,NOARRS;NO. OF ELEMENTS IN EACH OF THE ARRAYS WHICH ARE
C       OF THE SAME LENGTH,NOINEA; AND NO. OF ELEMENTS IN THE LAST ARRAY,
C       NOINLS
C
      NOONLN=64/(IFMTSZ+1)
      NOLINS=NOELEM/NOONLN
      ITST1=NOONLN*NOLINS
      IF(ITST1.LT.NOELEM) NOLINS=NOLINS+1
      NOARSM=NOLINS/15
      NOARRS=NOARSM
      ITST2=15*NOARSM
```

```
      IF(ITST2.LT.NOLINS) NOARRS=NOARRS+1
      NOINEA=NOONLN*15
      NOINLS=(NOELEM - (NOARSM*NOINEA) )
      NOPAIR=NOARSM/2
      NODD=NOARSM-2*NOPAIR
C
C       NEXT, CALCULATE THE INDICES OF THE INPUT ARRAY FOR
C       EQUIVALENCING.
C
      ID=-NOINEA+1
      DO 10 L=1,NOARRS
      ID=ID+NOINEA
      INDEX(L)=ID
   10 CONTINUE
C
C       NEXT, SET UP INDICES FOR ARRAY TYPE PRINTOUT IN
C       DIMENSIONING STATEMENT
C
      IF(NTYPE.EQ.1) I1=1
      IF(NTYPE.EQ.1) I2=3
      IF(NTYPE.EQ.2) I1=4
      IF(NTYPE.EQ.2) I2=6
      IF(NTYPE.EQ.3) I1=7
      IF(NTYPE.EQ.3) I2=9
      IF(NTYPE.EQ.4) I1=10
      IF(NTYPE.EQ.4) I2=12
C
      WRITE(6,2000)
      WRITE(7,2050)
      WRITE(6,2100) (TYPES(I),I=I1,I2),ARNAME,NOELEM
      WRITE(7,2150) (TYPES(I),I=I1,I2),ARNAME,NOELEM
C
C       DEBUG DATA
C
      WRITE(9,1000)NOONLN,NOLINS,NOARSM,NOARRS,NOINEA,NOINLS,
     1 IFMTSZ,NOELEM
 1000 FORMAT(1H1,'DEBUG DATA'/,1X,'NOONLN=',I5,2X,'NOLINS=',
     1 I5,2X,'NOARSM=',I5,2X,'NOARRS=',I5,2X,'NOINEA=',I5,
     2 2X,'NOINLS=',I5,2X,'IFMTSZ=',I5,2X,'NOELEM=',I5//)
      WRITE(9,1050) FORMS1,FORMS2,FORMS3,FORMS4
 1050 FORMAT(1HO,'FORMS ARRAYS'/,1X,8A4/,1X,7A4/,1X,32A4/,1X,
     1 3A4/,1X,32A4/,1X,2A4//,1X,'FOUT ARRAYS'/)
      WRITE(9,1100) FOUT1,FOUT2,FOUT3,FOUT4
 1100 FORMAT(1HO,20A4/,1X,20A4/,1X,20A4/,1X,20A4/)
C
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
C       GO TO SECTION FOR 'TYPE' OF DATA.
C
      GO TO (100,100,200,200),NTYPE
  100 CONTINUE
       NN=NOONLN
      NI=NOINEA-NOONLN*14 -1
      KK=NOINLS/NOONLN
      NIT=NOONLN*KK
      IF(NIT.EQ.NOINLS)KK=KK-1
      IF(NIT.EQ.NOINLS)NIT=NOONLN*KK
      NL=NOINLS-NIT-1
      IF(NL.LT.0)NL=0
      CALL INCORE(NN,FT1(13), 19,  1,  2,  0,  0)
      CALL INCORE(NI,FT1(45), 19,  1,  2,  0,  0)
      CALL INCORE(KK,FT3( 3), 19,  1,  2,  0,  0)
      CALL INCORE(NN,FT2(13), 19,  1,  2,  0,  0)
      CALL INCORE(NI,FT2(45), 19,  1,  2,  0,  0)
      CALL INCORE(KK,FT4( 3), 19,  1,  2,  0,  0)
      CALL INCORE(NN,FT3(13), 19,  1,  2,  0,  0)
      CALL INCORE(NL,FT3(45), 19,  1,  2,  0,  0)
      CALL INCORE(NN,FT4(13), 19,  1,  2,  0,  0)
      CALL INCORE(NL,FT4(45), 19,  1,  2,  0,  0)
C
      FOUT1(5)=FMT1(1)
      FOUT1(6)=FMT1(2)
      FOUT1(13)=FMT1(1)
      FOUT1(14)=FMT1(2)
      FOUT1(17)=FMT1(1)
      FOUT1(18)=FMT1(2)
      FOUT2(5)=FMT1(1)
      FOUT2(6)=FMT1(2)
      FOUT2(13)=FMT1(1)
      FOUT2(14)=FMT1(2)
      FOUT2(17)=FMT1(1)
      FOUT2(18)=FMT1(2)
      FOUT3(5)=FMT1(1)
      FOUT3(6)=FMT1(2)
      FOUT3(13)=FMT1(1)
      FOUT3(14)=FMT1(2)
      FOUT3(17)=FMT1(1)
      FOUT3(18)=FMT1(2)
      FOUT4(5)=FMT1(1)
      FOUT4(6)=FMT1(2)
      FOUT4(13)=FMT1(1)
      FOUT4(14)=FMT1(2)
```

```
      FOUT4(17)=FMT1(1)
      FOUT4(18)=FMT1(2)
C
C       DEBUG
C
      WRITE(9,1100) FOUT1,FOUT2,FOUT3,FOUT4
C
C
      IF(NOARRS.EQ.1) GO TO 160
C
      WRITE(6,FORMS1) (TYPES(I),I=I1,I2),ARNMSR(1),NOINEA
      WRITE(7,FORMS2) (TYPES(I),I=I1,I2),ARNMSR(1),NOINEA
      IF(NOARSM.GT.1) WRITE(6,FORMS3)(ARNMSR(I),NOINEA,I=2,NOARSM)
      IF(NOARSM.GT.1) WRITE(7,FORMS4)(ARNMSR(I),NOINEA,I=2,NOARSM)
      WRITE(6,2200) ARNMSR(NOARRS),NOINLS
      WRITE(7,2250) ARNMSR(NOARRS),NOINLS
      WRITE(6,2300)
      WRITE(7,2350)
      NOW=NOARSM
      IF(NODD.NE.0)NOW=NOW-1
      IF(NOW.GT.0) WRITE(6,2400) (ARNAME,INDEX(I),ARNMSR(I),I=1,NOW)
      IF(NOW.GT.0) WRITE(7,2450) (ARNAME,INDEX(I),ARNMSR(I),I=1,NOW)
      IF(NODD.NE.0)WRITE(6,2500)ARNAME,INDEX(NOARSM),ARNMSR(NOARSM),
     1 ARNAME,INDEX(NOARRS),ARNMSR(NOARRS)
      IF(NODD.NE.0)WRITE(7,2550)ARNAME,INDEX(NOARSM),ARNMSR(NOARSM),
     1 ARNAME,INDEX(NOARRS),ARNMSR(NOARRS)
      IF(NODD.EQ.0)WRITE(6,2600) ARNAME,INDEX(NOARRS),ARNMSR(NOARRS)
      IF(NODD.EQ.0) WRITE(7,2650) ARNAME,INDEX(NOARRS),ARNMSR(NOARRS)
C
C       HERE, REAL*4 AND REAL*8 DIFFER
C
      GO TO (120,140),NTYPE
  120 CONTINUE
      NOUSE1=-NOINEA+1
      DO 130 L=1,NOARSM
      WRITE(6,2700) ARNMSR(L)
      WRITE(7,2750) ARNMSR(L)
      NOUSE1=NOUSE1+NOINEA
      NOUSE2=NOUSE1+NOINEA-1
      WRITE(6,FOUT1) (ARR4(K),K=NOUSE1,NOUSE2)
      WRITE(7,FOUT2) (ARR4(K),K=NOUSE1,NOUSE2)
  130 CONTINUE
      WRITE(6,2700) ARNMSR(NOARRS)
      WRITE(7,2750) ARNMSR(NOARRS)
      NOUSE1=NOUSE1+NOINEA
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
      WRITE(6,FOUT3) (ARR4(L),L=NOUSE1,NOELEM)
      WRITE(7,FOUT4) (ARR4(L),L=NOUSE1,NOELEM)
      GO TO 300
  140 CONTINUE
      NOUSE1=-NOINEA+1
      DO 150 L=1,NOARSM
      WRITE(6,2700) ARNMSR(L)
      WRITE(7,2750) ARNMSR(L)
      NOUSE1=NOUSE1+NOINEA
      NOUSE2=NOUSE1+NOINEA-1
      WRITE(6,FOUT1) (ARR8(K),K=NOUSE1,NOUSE2)
      WRITE(7,FOUT2) (ARR8(K),K=NOUSE1,NOUSE2)
  150 CONTINUE
      WRITE(6,2700) ARNMSR(NOARRS)
      WRITE(7,2750) ARNMSR(NOARRS)
      NOUSE1=NOUSE1+NOINEA
      WRITE(6,FOUT3) (ARR8(L),L=NOUSE1,NOELEM)
      WRITE(7,FOUT4) (ARR8(L),L=NOUSE1,NOELEM)
      GO TO 300
C
C       ONE ARRAY!
C
  160 CONTINUE
      WRITE(6,2700) ARNAME
      WRITE(7,2750) ARNAME
      GO TO (170,180),NTYPE
  170 CONTINUE
      WRITE(6,FOUT3) (ARR4(L),L=1,NOELEM)
      WRITE(7,FOUT4) (ARR4(L),L=1,NOELEM)
      GO TO 300
  180 CONTINUE
      WRITE(6,FOUT3) (ARR8(L),L=1,NOELEM)
      WRITE(7,FOUT4) (ARR8(L),L=1,NOELEM)
      GO TO 300
C
C       INTEGER DATA PROCESSED HERE
C
C
  200 CONTINUE
       NN=NOONLN
      NI=NOINEA-NOONLN*14-1
      KK=NOINLS/NOONLN
      NIT=NOONLN*KK
      IF(NIT.EQ.NOINLS) KK=KK-1
      IF(NIT.EQ.NOINLS) NIT=NOONLN*KK
```

```
      NL=NOINLS-NIT-1
      IF(NL.LT.0)NL=0
      CALL INCORE(NN,FT1(13), 19,  1,  2,  0,  0)
      CALL INCORE(NI,FT1(45), 19,  1,  2,  0,  0)
      CALL INCORE(KK,FT3( 3), 19,  1,  2,  0,  0)
      CALL INCORE(NN,FT2(13), 19,  1,  2,  0,  0)
      CALL INCORE(NI,FT2(45), 19,  1,  2,  0,  0)
      CALL INCORE(KK,FT4( 3), 19,  1,  2,  0,  0)
      CALL INCORE(NN,FT3(13), 19,  1,  2,  0,  0)
      CALL INCORE(NL,FT3(45), 19,  1,  2,  0,  0)
      CALL INCORE(NN,FT4(13), 19,  1,  2,  0,  0)
      CALL INCORE(NL,FT4(45), 19,  1,  2,  0,  0)
C
      FOUT1(5)=FMT1(1)
      FOUT1(6)=FMT1(2)
      FOUT1(13)=FMT1(1)
      FOUT1(14)=FMT1(2)
      FOUT1(17)=FMT1(1)
      FOUT1(18)=FMT1(2)
      FOUT2(5)=FMT1(1)
      FOUT2(6)=FMT1(2)
      FOUT2(13)=FMT1(1)
      FOUT2(14)=FMT1(2)
      FOUT2(17)=FMT1(1)
      FOUT2(18)=FMT1(2)
      FOUT3(5)=FMT1(1)
      FOUT3(6)=FMT1(2)
      FOUT3(13)=FMT1(1)
      FOUT3(14)=FMT1(2)
      FOUT3(17)=FMT1(1)
      FOUT3(18)=FMT1(2)
      FOUT4(5)=FMT1(1)
      FOUT4(6)=FMT1(2)
      FOUT4(13)=FMT1(1)
      FOUT4(14)=FMT1(2)
      FOUT4(17)=FMT1(1)
      FOUT4(18)=FMT1(2)
C
C       DEBUG
C
      WRITE(9,1100) FOUT1,FOUT2,FOUT3,FOUT4
C
C
      IF(NOARRS.EQ.1) GO TO 260
C
```


REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
      WRITE(6,FORMS1) (TYPES(I),I=I1,I2),ARNMSI(1),NOINEA
      WRITE(7,FORMS2) (TYPES(I),I=I1,I2),ARNMSI(1),NOINEA
      IF(NOARSM.GT.1) WRITE(6,FORMS3)(ARNMSI(I),NOINEA,I=2,NOARSM)
      IF(NOARSM.GT.1) WRITE(7,FORMS4)(ARNMSI(I),NOINEA,I=2,NOARSM)
      WRITE(6,2200) ARNMSI(NOARRS),NOINLS
      WRITE(7,2250) ARNMSI(NOARRS),NOINLS
      WRITE(6,2300)
      WRITE(7,2350)
      NOW=NOARSM
      IF(NODD.NE.0) NOW=NOW-1
      IF(NOW.GT.0) WRITE(6,2400) (ARNAME,INDEX(I),ARNMSI(I),I=1,NOW)
      IF(NOW.GT.0) WRITE(7,2450) (ARNAME,INDEX(I),ARNMSI(I),I=1,NOW)
      IF(NODD.NE.0) WRITE(6,2500)ARNAME,INDEX(NOARSM),ARNMSI(NOARSM),
     1 ARNAME,INDEX(NOARRS),ARNMSI(NOARRS)
      IF(NODD.NE.0) WRITE(7,2550)ARNAME,INDEX(NOARSM),ARNMSI(NOARSM),
     1 ARNAME,INDEX(NOARRS),ARNMSI(NOARRS)
      IF(NODD.EQ.0) WRITE(6,2600) ARNAME,INDEX(NOARRS),ARNMSI(NOARRS)
      IF(NODD.EQ.0) WRITE(7,2650) ARNAME,INDEX(NOARRS),ARNMSI(NOARRS)
C
C       HERE, INTEGER*2 AND INTEGER*4 DIFFER
C
      NTY=NTYPE-2
      GO TO (220,240),NTY
  220 CONTINUE
      NOUSE1=-NOINEA+1
      DO 230 L=1,NOARSM
      WRITE(6,2700) ARNMSI(L)
      WRITE(7,2750) ARNMSI(L)
      NOUSE1=NOUSE1+NOINEA
      NOUSE2=NOUSE1+NOINEA-1
      WRITE(6,FOUT1) (INTAR2(K),K=NOUSE1,NOUSE2)
      WRITE(7,FOUT2) (INTAR2(K),K=NOUSE1,NOUSE2)
  230 CONTINUE
      WRITE(6,2700) ARNMSI(NOARRS)
      WRITE(7,2750) ARNMSI(NOARRS)
      NOUSE1=NOUSE1+NOINEA
      WRITE(6,FOUT3) (INTAR2(L),L=NOUSE1,NOELEM)
      WRITE(7,FOUT4) (INTAR2(L),L=NOUSE1,NOELEM)
      GO TO 300
  240 CONTINUE
      NOUSE1=-NOINEA+1
      DO 250 L=1,NOARSM
      WRITE(6,2700) ARNMSI(L)
      WRITE(7,2750) ARNMSI(L)
      NOUSE1=NOUSE1+NOINEA
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
      NOUSE2=NOUSE1+NOINEA-1
      WRITE(6,FOUT1) (INTAR4(K),K=NOUSE1,NOUSE2)
      WRITE(7,FOUT2) (INTAR4(K),K=NOUSE1,NOUSE2)
  250 CONTINUE
      WRITE(6,2700) ARNMSI(NOARRS)
      WRITE(7,2750) ARNMSI(NOARRS)
      NOUSE1=NOUSE1+NOINEA
      WRITE(6,FOUT3) (INTAR4(L),L=NOUSE1,NOELEM)
      WRITE(7,FOUT4) (INTAR4(L),L=NOUSE1,NOELEM)
      GO TO 300
C
C       ONE ARRAY!
C
  260 CONTINUE
      WRITE(6,2700) ARNAME
      WRITE(7,2750) ARNAME
      NTY=NTYPE-2
      GO TO (270,280),NTY
  270 CONTINUE
      WRITE(6,FOUT3) (INTAR2(L),L=1,NOELEM)
      WRITE(7,FOUT4) (INTAR2(L),L=1,NOELEM)
      GO TO 300
  280 CONTINUE
      WRITE(6,FOUT3) (INTAR4(L),L=1,NOELEM)
      WRITE(7,FOUT4) (INTAR4(L),L=1,NOELEM)
  300 CONTINUE
      WRITE(6,2800)
      WRITE(7,2850)
      RETURN
      END
```

APPENDIX H - QUICK PROGRAM LISTINGS, INPUT, AND JCL OUTPUT

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

11JUL74 10.04.45 JCLE192R

```
00000100  // EXEC LINKGO,REGION.GO=400K
00000120  //LINK.OBJECT DD *
00000140   INCLUDE LOADLIB(ZBMJMR92)
00000200  //*     UNIT 6 IS THE DEBUG AND GENERAL MESSAGE UNIT ON PRINTER
00000300  //*      UNIT 9 CONTAINS THE LISTING OF THE TAPE CONTENTS
00000400  //*       TP192 IS THE INPUT DATA TAPE
00000500  //GO.FT06F001 DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=7265),
00000600  // SPACE=(CYL,(7,3))
00000700  //GO.FT09F001 DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=7265),
00000800  // SPACE=(CYL,(15,5))
00000900  //GO.TP192 DD UNIT=2400-9,DCB=(RECFM=U,BLKSIZE=32000),LABEL=(,BLP),
00001000  // VOL=SER=L0045
00001100  //GO.SYSUDUMP DD SYSOUT=A
00001200  //GO.DATA5 DD *

*** END OF MEMBER ***      14 RECORDS PROCESSED.
```

```
C  THIS PROGRAM WAS WRITTEN BY F. MILASI OF CSC FOR TASK 092,
C SKYLAB, UNDER CONTRACT NAS-5-11999 IN APRIL, 1974. IT
C UTILIZES SUBROUTINES REFLE1,CLUTHK,REFLE2,DELTA1,WRITT,SPLINE.
C  USING THE READ PROGRAM ALREADY DEVELOPED FOR THE S192 DATA,
C THIS PROGRAM WILL PICK OFF POINTS IN THE EXISTING DATA TAPE,
C DO A MINIMAL AMOUNT OF PROCESSING AND FINALLY PRINT OUT THE
C RESULTS.
C  THE MANNER IN WHICH THE POINTS WILL BE PICKED WILL BE AS
C FOLLOWS: EVERY THIRTIETH POINT ON A SCAN LINE, AND EVERY
C THIRTIETH SCAN LINE WILL BE STORED AND PROCESS.
C*****************************************************************
C*****************************************************************
      COMMON/VALUES/R1(35,67),T1(35,67),RR(35,67),Y1(35,67),
     1X1(35,67),D1(35,67),E(35,67),E1(35,67),TBB(35,67),R4(35,67),
     1LC,LR,LC2,LR2,LR3,LC3,LR4,LC4,PLOT1,PLOT2,PLOT3,PLOT4,PLOT5
      DIMENSION SCAN(1301),ISCAN(1301),NVAL(3),A(35,67),B(35,67),
     1PIXEL(1240),C(35,67),D(35,67)
      INTEGER*2 ISCAN2(2602),DEG,AMIN,QSEC
      NAMELIST/INPUT/ NCHAN1, NCHAN2,NUMPIT,NUMLIN,IPIXEL,NCHAN3,
     1NCHAN4,PLOT1,PLOT2,PLOT3,PLOT4,PLOT5,NUMPLT
      LOGICAL*4 PLOT1,PLOT2,PLOT3,PLOT4,PLOT5
      DATA NCHAN1/0/
      DATA NCHAN2/0/
      DATA NCHAN3/0/
      DATA NCHAN4/0/
      DATA IPIXEL/0/
      DATA IPXODD/37/
      DATA IPXEVN/19/
      DATA LASODD/1050/
      DATA LASEVN/1068/
      DATA ILAST/0/
      DATA NUMPIT/30/
      DATA NUMLIN/30/
      DATA NVAL/0,0,0/
      DATA RADI/.017453/
      EQUIVALENCE (NVAL(1),NCOND), (NVAL(2),NHOUS),(NVAL(3),NSCAN)
      EQUIVALENCE (SCAN(1),ISCAN(1),ISCAN2(1))
      EQUIVALENCE (ISCAN2(50),DEG),(ISCAN2(51),AMIN),
     1(ISCAN2(52),QSEC)

      EQUIVALENCE (PIXEL(1),SCAN(62))
      DATA NCOUNT/0/
      DATA NCOU1/0/
      DATA NCOU2/0/
      DATA NCOU3/0/
```


REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
      -DATA NCOU4/0/
C*****************************************************************************
C  CHANNELS 15 AND 16 ON THE TAPE ARE NOT ACTIVE. AS THERE ARE
C 22 ON THE TAPE, THERE ARE 20 ACTIVE. THE REASON FOR THIS WAY
C WAS THATT THE 'CHANNEL' NO. GIVEN THEM BY HOUSTON DOES NOT
C CORRESPOND TO THE 'CHANNEL NO.' BY WHICH THE SCIENTISTS
C REFERENCE THEM, (ACCORDING TO DR. CURRAN AND CORD OF GSFC).
C IT THEREFORE SEEMED USELESS TO RENUMBER THEM HERE BY
C HOUSTON'S SCHEME.
C  ALSO 19TH ACTIVE CHANNEL WILL NOT BE USED BECAUSE THE
C IRRADIANCE DATA ASSOCIATED WITH THE WAVELENGTHS ARE EXTREMELY
C SMALL AND INSIGNIFICANT.
C*****************************************************************************
      NUMPLT=0
      PLOT1=.FALSE.
      PLOT2=.FALSE.
      PLOT3=.FALSE.
      PLOT4=.FALSE.
      PLOT5=.FALSE.

      READ(5, INPUT)
      IF(NCHAN1.GT.22.OR.NCHAN2.GT.22)GO TO 95
      IF(NCHAN3.GT.22.OR.NCHAN4.GT.22)GO TO 95
      IF(NCHAN1.EQ.15.OR.NCHAN1.EQ.16)GO TO 90
      IF(NCHAN2.EQ.15.OR.NCHAN2.EQ.16)GO TO 90
      IF(NCHAN3.EQ.15.OR.NCHAN3.EQ.16)GO TO 90
      IF(NCHAN4.EQ.15.OR.NCHAN4.EQ.16)GO TO 90
      IF(NUMPLT.NE.5)GO TO 5
      PLOT1=.TRUE.
      PLOT2=.TRUE.
      PLOT3=.TRUE.
      PLOT4=.TRUE.
      PLOT5=.TRUE.
    5 LR3=0
      LC3=0
      LASTR1=0
      LASTR2=0
      LASTC1=0
      LASTC2=0
      LR4=0
      LC4=0
   10 CONTINUE
      NCOUNT=NCOUNT+1
      IPIXEL=IPXODD
      ILAST=LASODD
```

```
      IF(MOD(NCOUNT,2).EQ.0)IPIXEL=IPXEVN
      IF(MOD(NCOUNT,2).EQ.0)ILAST=LASEVN
      CALL RD192(SCAN,NVAL)
      IF(NCOND.EQ.1)GO TO 70
      IF(NCOND.EQ.2)GO TO 80
      IF(ISCAN2(1).EQ.NCHAN1)GO TO 20
      IF(ISCAN2(1).EQ.NCHAN2)GO TO 30
      IF(ISCAN2(1).EQ.NCHAN3)GO TO 40
      IF(ISCAN2(1).EQ.NCHAN4)GO TO 50
      GO TO 10
   20 CONTINUE
C ***************************************************************************
C     READING OF FRIST CHANNEL OR 'NCHAN1' THAT WAS REQUESTED
C BY USER.....'NCHAN1' WILL BE STORED IN ARRAY 'A'.
C***************************************************************************
      NCOU1=NCOU1+1
      LINEN1=ISCAN2(4)
      NROWM=MOD(LINEN1,NUMLIN)
      IF(NROWM.NE.1)GO TO 10
      NROW1=LINEN1/NUMLIN+1
      IF(LASTRW.GT.67)GO TO 60
      LASTRW=NROW1
      NCOL1=0
      DO 25 N=IPIXEL,ILAST,NUMPIT
      NCOL1=NCOL1+1
      LASTCL=NCOL1
      IF(NCOL1.GT.35) GO TO 25
      A(NCOL1,NROW1)=PIXEL(N)
   25 CONTINUE
      LR=LASTRW
      LC=LASTCL
      IF(LASTRW.EQ.67)LR=LR-2
      IF(LASTCL.EQ.35)LC=LC-2
      IF(LASTCL.EQ.34)LC=LC-1
      ANGG=DEG+AMIN/60.+OSEC/240.
      ANG=ANGG*RADI
      GO TO 10
C***************************************************************************
C     READING OF SECOND CHANNEL OR 'NCHAN2' THAT WAS REQUESTED
C BY USER.....'NCHAN2' WILL BE STORED IN ARRAY 'B'.
C***************************************************************************
   30 LINEN2=ISCAN2(4)
      NCOU2=NCOU2+1
      NROWM=MOD(LINEN2,NUMLIN)
      IF(NROWM.NE.1)GO TO 10
```


REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

```
      NROW2=LINEN2/NUMLIN+1
      IF(LASTR2.GT.67)GO TO 60
      LASTR2=NROW2
      NCOL2=0
      DO 35 N=IPIXEL,ILAST,NUMPIT
      NCOL2=NCOL2+1
      LASTC2=NCOL2
      IF(NCOL2.GT.35)GO TO 35
      B(NCOL2,NROW2)=PIXEL(N)
   35 CONTINUE
       LR2=LASTR2
      LC2=LASTC2
      IF(LASTC2.EQ.35)LC2=LC2-2
      IF(LASTC2.EQ.34)LC2=LC2-1
      IF(LASTR2.EQ.67)LR2=LR2-2
      ANG2=DEG+AMIN/60.+QSEC/240.
      ANG2=ANG2*RADI
      IF(LINEN2.EQ.1981)GO TO 60
      GO TO 10
C*************************************************************************
C     READING OF THIRD CHANNEL OR 'NCHAN3' THAT WAS REQUESTED
C BY USER.......'NCHAN3' WILL BE STORED IN ARRAY 'C'.
C*************************************************************************
   40 CONTINUE
      NCOU3=NCOU3+1
      LINEN3=ISCAN2(4)
      NROWM=MOD(LINEN3,NUMLIN)
      IF(NROWM.NE.1)GO TO 10
      NROW3=LINEN3/NUMLIN+1
      IF(LR3.GT.67)GO TO 60
      LR3=NROW3
      NCOL3=0
      DO 45 N=IPIXEL,ILAST,NUMPIT
      NCOL3=NCOL3+1
      LC3=NCOL3
      IF(NCOL3.GT.35)GO TO 45
      C(NCOL3,NROW3)=PIXEL(N)
   45 CONTINUE
      IF(LR3.EQ.67)LR3=LR3-2
      IF(LC3.EQ.34)LC3=LC3-1
      IF(LC3.EQ.35)LC3=LC3-2
      GO TO 10
C*************************************************************************
C     READING OF FOURTH CHANNEL OR 'NCHAN4' THAT WAS REQUESTED
C BY USER......'NCHAN4' WILL BE STORED IN ARRAY 'D'
```

```
C**********************************************************************
   50 CONTINUE
      NCOU4=NCOU4+1
      LINEN4=ISCAN2(4)
      NROWM=MOD(LINEN4,NUMLIN)
      IF(NROWM.NE.1)GO TO 10
      NROW4=LINEN4/NUMLIN+1
      IF(LR4.GT.67)GO TO 60
      LR4=NROW4
      NCOL4=0
      DO 55 N=IPIXEL,ILAST,NUMPIT
      NCOL4=NCOL4+1
      LC4=NCOL4
      IF(NCOL4.GT.35)GO TO 55
      D(NCOL4,NROW4)=PIXEL(N)
   55 CONTINUE
      IF(LR4.EQ.67)LR4=LR4-2
      IF(LC4.EQ.35)LC4=LC4-2
      IF(LC4.EQ.34)LC4=LC4-1
      ANG4=DEG+AMIN/60.+0SEC/240.
      ANG4=ANG4*RADI
      GO TO 10
   60 CONTINUE
      WRITE(6,3000)
      IF(NUMPLT.EQ.0)GO TO 63
      CALL NOSTAE
      CALL PLOTST(1001,4)
   63 IF(NCHAN3.EQ.0)GO TO 65
      CALL TEMBRH(NCHAN3,C,NUMPLT)
      IF(NCHAN1.EQ.0)GO TO 100
   65 CALL REFLE1(A,ANG,NCHAN1,B,ANG2,NCHAN2,D,ANG4,NCHAN4,NUMPLT)
      STOP

   70 WRITE(6,1000)
      STOP
   80 WRITE(6,2000)NCOUNT,NCOU1,NCOU2,NCOU3,NCOU4
      WRITE(6, INPUT)
      IF(NUMPLT.EQ.0)GO TO 83
      CALL NOSTAE
      CALL PLOTST(1001,4)
   83 IF(NCHAN3.EQ.0)GO TO 85
      CALL TEMBRH(NCHAN3,C,NUMPLT)
   85 CALL REFLE1(A,ANG,NCHAN1,B,ANG2,NCHAN2,D,ANG4,NCHAN4,NUMPLT)
      STOP
   90 WRITE(6,4000)NCHAN1,NCHAN2,NCHAN3,NCHAN4
```


REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
      STOP
   95 WRITE(6,6000)NCHAN1,NCHAN2,NCHAN3,NCHAN4
  100 WRITE(6,7000)NCHAN1,NCHAN2,NCHAN3,NCHAN4
      STOP
 1000 FORMAT(1HO,10X,'TAPE ERROR')
 2000 FORMAT(1HO,10X,'END OF FILE',5X,I6,5X,I4,5X,I4,5X,I4,5X,I4)
 3000 FORMAT(1HO,10X,'CALL BEFORE END OF FILE')
 4000 FORMAT(1HO,10X,'PLEASE CHECK CHANNEL THAT YOU REQUESTED,
     1CHANNELS 15,16,    ARE NOT ACTIVE',5X,'NCHAN1=',I3,5X,
     2'NCHANE2=',I3,5X,'NCHAN3',I3,5X,'NCHAN4',I3)
 6000 FORMAT(1HO,5X,'PLEASE CHECK INPUT CHANNEL NUMBER...',
     1'NCHAN1=',I3,' NCHAN2=',I3,'NCHAN3=',I3,'NCHAN4=',I3,
     25X,'YOU HAVE REQUESTED FOR A CHANNEL GREATER THAN 22',
     3'(THERE ARE ONLY 22 ACTIVE CHANNELS ON S192 TAPE)')
 7000 FORMAT(1HO,10X,'ONLY ONE CHANNEL WAS REQUESTED,NNCHAN3',5X,
     1'NCHAN1=',I3,'NCHAN2=',I3,'NCHAN3=',I3,'NCHAN4=',I3)
      END
      SUBROUTINE TEMBRH(NCHAN3,C,NUMPLT)
      COMMON/VALUES/R1(35,67),T1(35,67),RR(35,67),Y1(35,67),
     1X1(35,67),D1(35,67),E(35,67),E1(35,67),TBB(35,67),R4(35,67),
     1LC,LR,LC2,LR2,LR3,LC3,LR4,LC4,PLOT1,PLOT2,PLOT3,PLOT4,PLOT5
      DIMENSION CHANNL(21),C(35,67)
      LOGICAL*4 PLOT1,PLOT2,PLOT3,PLOT4,PLOT5
      DATA CONE/3.7413E+04/
      DATA CTWO/1.4388E+04/
      DATA PI/3.1416/
      DATA CHANNL/.540,.540,.585,.585,.645,.645,.720,.720,
     1.830,.830,1.650,1.650,2.225,2.225,.000,.000,12.500,
     2.485,1.005,1.140,11.350/
      DO 10 I=1,21
      WAVET=CHANNL(I)
      IF(NCHAN3.EQ.I)GO TO 15
   10 CONTINUE
      WRITE(6,1000)NCHAN3
      STOP
   15 TTT=PI*WAVET**5
      DO 20 I=1,LC3
      DO 30 J=1,LR3
      TBB(I,J)=0.
      IF(C(I,J).LE.0.)GO TO 30
      TAB=1.+(CONE/(TTT*C(I,J)))
      TAA=WAVET*ALOG(TAB)
      TBB(I,J)=CTWO/TAA
   30 CONTINUE
   20 CONTINUE
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
      CALL WRITT(6)
      IF(PLOT1)CALL CONTOU(TBB,LR3,LC3,NUMPLT,1)
      RETURN
 1000 FORMAT(1H0,5X,'NCHAN3 DOES NOT AGREE WITH WAVELENGTH ,
     1CHANNEL',5X,'NCHAN3=',I3)
      END
      SUBROUTINE REFLE1(A,ANG,NCHAN1,B,ANG2,NCHAN2,D,ANG4,NCHAN4,NUMPLT
C
C  THIS SUBROUTINE WAS WRITTEN BY F. MILASI OF CSC FOR TASK 092,
C SKYLAB, UNDER CONTRACT NAS-5-11999 IN APRIL, 1974. SUBROUTINE
C REFLE1 TAKES MATRIX 'A', WHICH WAS PASSED FROM THE MAIN
C PROGRAM AND  WAS FORMED FROM NCHAN1.
C  THIS WILL BE TRANSFORMED INTO REFLECTANCES BY THE MULTIPLICATION
C OF EACH ELEMENT BY THE FACTOR (PI/FO*COS(ANG)).     THE
C RESULTING MATRIX 'R1' WILL BE MULTIPLIED BY 1000. (TO ALLOW
C FIXED POINT FORMAT) AND WILL BE PRINTED OUT BY SUBROUTINE WRITT.
C**********************************************************************
C**********************************************************************
      COMMON/VALUES/R1(35,67),T1(35,67),RR(35,67),Y1(35,67),
     1X1(35,67),D1(35,67),E(35,67),E1(35,67),TBB(35,67),R4(35,67),
     1LC,LR,LC2,LR2,LR3,LC3,LR4,LC4,PLT01,PLOT2,PLOT3,PLOT4,PLOT5
      DIMENSION A(35,67),B(35,67),D(35,67)
      LOGICAL*4 PLOT1,PLOT2,PLOT3,PLOT4
C
C**********************************************************************
C  ARRAY FF IS THE ARRAY OF IRRADIANCE DATA ASSOCIATED WITH THE
C AVERAGE WAVELENGTH FOR EACH ACTIVE CHANNED. THESE VALUES WERE
C DERIVED THRU GAUSSIAN 'AVERAGING' OF DR. THEKAKARA IRRADIANCE
C DATA USING  CNTHGS92.
C      NOTE: THE FF IRRADIANCE DATA GIVEN FOR THE 15TH AND 19TH
C ACTIVE CHANNELS ARE MERELY DUMMY VALUES. SINCE IRRADIANCE
C DATA ASSOCIATED WITH THE WAVELENGTHS REPUSENTED BY THOSE
C CHANNELS ARE EXTREMELY SMALL AND INSIGNIFICANT, NO
C CALCULAT USING THEM WOULD BE SCIENTIFICALLY MEANINGFUL,
C (ACCORDING TO DR. CURRAN OF GSFC). SO, AS THESE CHANNELS
C WOULD NEVER BE USED, DERVATION OF ACCURATE IRRADIANCE VALUES
C SEEMED POINTLESS.
C**********************************************************************
C
      REAL*8 FF(22),FO
      DATA FF/0.1784488D0,0.1784488D0,0.1704564D0,0.1704564D0,
     10.1526697D0,0.1526697D0,0.1310974D0,0.1310974D0,0.1034361D0,
     10.1034361D0,0.0220749D0,0.0220749D0,0.0081098D0,0.0081098D0,
     10.0000000D0,0.0000000D0,0.0000000D0,0.1999891D0,0.0762900D0,
     10.0549257D0,0.0000000D0,0.1723639D0/
```

```
      DATA PI/3.1416/
      DO 5 M=1,22
      FO=FF(M)
      IF(F.EQ.NCHAN1)GO TO 7
    5 CONTINUE
      WRITE(6,1000)NCHAN1
      STOP
    7 CONTINUE
      CA=COS(ANG)
      F=FO*CA
      FAC=PI/F
      DO 10 I=1,LC
      DO 20 J=1,LR
      R1(I,J)=0.0
      IF(A(I,J).LE.0.)GO TO 20
      R1(I,J)=A(I,J)*FAC
   20 CONTINUE
   10 CONTINUE
      DO 30 L=1,LC
      DO 40 N=1,LR
      RR(L,N)=R1(L,N)*1000.
   40 CONTINUE
   30 CONTINUE
      CALL WRITT(1)
      IF(PLOT2)CALL CONTOU(RR,LR,LC,NUMPLT,2)
      IF(NCHAN2.NE.0)GO TO 45
      WRITE(6,2000)NCHAN1,NCHAN2,NCHAN4
      STOP
   45 NUM=0
      CALL CLUTHK(NUM,NUMPLT)
      CALL REFLE2(B,ANG2,NCHAN2,FF,D,ANG4,NCHAN4,NUMPLT)
 1000 FORMAT(1HO,10X,'NCHAN1 DOES NOT AGREE WITH FF ARRAY',
     15X,'NCHAN1=',I3)
 2000 FORMAT(1HO,'PROGRAM STOP-----ONLY ONE CHANNEL WAS REQUESTED,
     1   NCHAN1=',I3,' NCHAN2=',I2,' NCHAN4=',I2)
      RETURN
      END
      SUBROUTINE REFLE2(B,ANG2,NCHAN2,FF,D,ANG4,NCHAN4,NUMPLT)
C
C   THIS SUBROUTINE WAS WRITTEN BY F. MILASI OF CSC FOR TASK 092,
C SKYLAB, UNDER CONTRACT NAS-5-11999 IN APRIL, 1974. SUBROUTINE
C REFLE2 TAKES MATRIX 'B' WHICH WAS PASSED FROM THE MAIN PROGRAM,
C AND WAS FORMED FROM NCHAN2. IT WILL BE TRANSFORMED INTO REFLECTANCES
C BY A SIMILAR PROCEDURE AS REFLE1 (EXCEPT THE QUANTITY FF FOR THIS
C CHANNEL IS DIFFERENT). THUS A REFLECTANCE MATRIX 'R2' WILL BE FORMED.
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
C A NEW MATRIX 'X1' WILL BE FORMED BY THE RELATIONSHIP X1=R2/R1
C WHERE R1 IS THE REFLECTANCE MATRIX OF SUBROUTINE REFLE1 BEFORE
C MULTIPLICATION BY 1000.
C*****************************************************************
C*****************************************************************
      COMMON/VALUES/R1(35,67),T1(35,67),RR(35,67),Y1(35,67),
     1X1(35,67),D1(35,67),E(35,67),E1(35,67),TBB(35,67),R4(35,67),
     1LC,LR,LC2,LR2,LR3,LC3,LR4,LC4,PLT01,PLOT2,PLOT3,PLOT4,PLOT5
      DIMENSION R(35,67),R2(35,67),D(35,67)
      LOGICAL*4 PLOT1,PLOT2,PLOT3,PLOT4,PLOT5
      REAL*8 FF(22),F0
      DATA PI/3.1416/
      DO 5 M=1,22
      F0=FF(M)
      IF(M.EQ.NCHAN2)GO TO 7
    5 CONTINUE
      WRITE(6,1000)NCHAN2
      STOP
    7 CONTINUE
      CA=COS(ANG2)
      F=F0*CA
      FAC=PI/F
      DO 10 I=1,LC2
      DO 20 N=1,LR2
      R2(I,N)=R(I,N)*FAC
   20 CONTINUE
   10 CONTINUE
      IF(NCHAN4.EQ.0)GO TO 25
      CALL REFLE3(FF,ANG4,NCHAN4,D,R2,NUMPLT)
   25 DO 30 J=1,LC2
      DO 40 K=1,LR2
      X1(J,K)=-.99
      IF(R1(J,K).LE.0..OR.R2(J,K).LE.0.)GO TO 40
      X1(J,K)=R2(J,K)/R1(J,K)
   40 CONTINUE
   30 CONTINUE
      DO 50 L=1,LC2
      DO 60 M=1,LR2
      RR(L,M)=X1(L,M)*100.
   60 CONTINUE
   50 CONTINUE
      CALL WRITT(3)
      NUM=1
      CALL CLUTHK(NUM,NUMPLT)
      DO 70 I=1,LC2
```

```
      DO 80 K=1,LR2
      D1(I,K)=0.0
      IF(Y1(I,K).EQ.0.0.OR.X1(I,K).EQ.-0.99)GO TO 80
      D1(I,K)=X1(I,K)-Y1(I,K)
   80 CONTINUE
   70 CONTINUE
      DO 90 J=1,LC2
      DO 100 K=1,LR2
      RR(J,K)=D1(J,K)*100.
  100 CONTINUE
   90 CONTINUE
      CALL WRITT(4)
      DO 110 L=1,LC2
      DO 120 M=1,LR2
      IF(T1(L,M).LE.R.)RR(L,M)=0.0
  120 CONTINUE
  110 CONTINUE
      IF(PLOT4)CALL CONTOU(RR,LR2,LC2,NUMPLT,4)
      NUMM=0
      CALL DELTA1(NUMM)
      RETURN
 1000 FORMAT(1HO,10X,'NCHAN2 DOES NOT AGREE WITH FF ARRAY',
     15X,'NCHAN2=',I3)
      END
      SUBROUTINE REFLE3(FF,ANG4,NCHAN4,D,R2,NUMPLT)
      COMMON/VALUES/R1(35,67),T1(35,67),RR(35,67),Y1(35,67),
     1X1(35,67),D1(35,67),E(35,67),E1(35,67),TBB(35,67),R4(35,67),
     2LC,LR,LC2,LR2,LR3,LC3,LR4,LC4,PLOT1,PLOT2,PLOT3,PLOT4,PLOT5
      REAL*8 FF(22),FO
      DIMENSION D(35,67),R2(35,67)
      LOGICAL*4 PLOT1,PLOT2,PLOT3,PLOT4,PLOT5
      DATA PI/3.1416/
      DO 10 M=1,22
      FO=FF(M)
      IF(M.EQ.NCHAN4)GO TO 15
   10 CONTINUE
      WRITE(6,1000)NCHAN4
      STOP
   15 CONTINUE
      CA=COS(ANG4)
      F=FO*CA
      FAC=PI/F
      DO 20 I=1,LC4
      DO 30 N=1,LR4
      R4(I,N)=D(I,N)*FAC
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
 30 CONTINUE
 20 CONTINUE
    DO 40 J=1,LC4
    DO 50 K=1,LR4
    X1(J,K)=-.99
    IF(R2(J,K).LE.0..OR.R4(J,K).LE.0.)GO TO 50
    X1(J,K)=R2(J,K)/R4(J,K)
 50 CONTINUE
 40 CONTINUE
    DO 60 L=1,LC4
    DO 70 M=1,LR4
     RR(L,M)=X1(L,M)*100.
 70 CONTINUE
 60 CONTINUE
    CALL WRITT(7)
    NUM=2
    CALL CLUTHK(NUM)
    DO 80 I=1,LC4
    DO 90 K=1,LR4
    D1(I,K)=0.0
    IF(Y1(I,K).LE.0.0.OR.X1(I,K).EQ.-.99)GO TO 90
     D1(I,K)=X1(I,K)-Y1(I,K)
 90 CONTINUE
 80 CONTINUE
    DO 100 J=1,LC4
    DO 110 K=1,LR4
     RR(J,K)=D1(J,K)*100.
110 CONTINUE
100 CONTINUE
    CALL WRITT(8)
    DO 120 L=1,LC4
    DO 130 M=1,LR4
    IF(T1(L,M).LE.8.)RR(L,M)=0.0
130 CONTINUE
120 CONTINUE
    IF(PLOT5)CALL CONTOU(RR,LR4,LC4,NUMPLT,5)
    NUMM=1
    CALL DELTA1(NUMM)
    RETURN
1000 FORMAT(1H0,10X,'NCHAN4 DOES NOT AGREE WITH FF ARRAY',5X,
   1'NCHAN4='I3)
    END
    SUBROUTINE CLUTHK(NUM,NUMPLT)
    COMMON/VALUES/R1(35,67),T1(35,67),RR(35,67),Y1(35,67),
   1X1(35,67),D1(35,67),E(35,67),E1(35,67),TBB(35,67),R4(35,67),
```

```
     1LC,LR,LC2,LR2,LR3,LC3,LR4,LC4,PLOT1,PLOT2,PLOT3,PLOT4,PLOT5
      DIMENSION X(10),Y(10),X0(19),Y0(19),F(19),U(19),XF(19),YF(19)
      LOGICAL*4 PLOT1,PLOT2,PLOT3,PLOT4,PLOT5
      REAL*8 S,X,Y,TX,VL,VU,F,U,X0,Y0,XF,YF
      DATA X/0.,0.038,0.092,0.248,0.422,0.607,0.759,0.848,
     10.892,0.920/
      DATA Y/0.,1.,2.,5.,10.,20.,50.,100.,200.,500./
      DATA X0/5.,6.,7.,8.,9.,10.,20.,30.,40.,50.,60.,70.,80.,90.,100.,
     1200.,300.,400.,500./
      DATA Y0/0.963,0.933,0.908,0.887,0.865,0.850,0.720,0.643,
     10.601,0.575,0.558,0.542,0.534,0.523,0.520,0.493,0.482,
     20.480,0.480/
      DATA XF/5.,6.,7.,8.,9.,10.,20.,30.,40.,50.,60.,70.,80.,90.,100.,
     1200.,300.,400.,500./
      DATA YF/1.04,.97,.917,.905,.880,.857,.719,.648,.612,
     2.590,.576,.567,.560,.555,.55,.534,.530,.530,.530/
      IF(NUM.EQ.0)GO TO 5
      IF(NUM.EQ.1)GO TO 40
      IF(NUM.EQ.2)GO TO 70
    5 N=10
      IL=3
      IU=3
      VL=0.
      VU=0.
      CALL SPLINE(S,X,Y,N,1,TX,IL,VL,IU,VU,F,U)
      DO 10 L=1,LC
      DO 20 M=1,LR

      S=R1(L,M)
      TX=0.0
       IF(S.LE.0.)GO TO 15
      CALL SPLINE(S,X,Y,N,2,TX,IL,VL,IU,VU,F,U)
   15 T1(L,M)=TX
   20 CONTINUE
   10 CONTINUE
      CALL WRITT(2)
      IF(PLOT3)CALL CONTOU(T1,LR,LC,NUMPLT,3)
      RETURN
   40 N=19
      IU=2
      IL=3
      VL=0.
      VU=0.
      CALL SPLINE(S,X0,Y0,N,1,TX,IL,VL,IU,VU,F,U)
      DO 50 I=1,LC2
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
      DO 60 K=1,LR2
      S=T1(I,K)
      TX=0.0
      IF(S.LE.0.)GO TO 55
      CALL SPLINE(S,XO,YO,N,2,TX,IL,VL,IU,VU,F,U)
   55 Y1(I,K)=TX
   60 CONTINUE
   50 CONTINUE
      RETURN
   70 N=19
      IU=2
      IL=3
      VL=0.
      VU=0.
      CALL SPLINE(S,XF,YF,N,1,TX,IL,VL,IU,VU,F,U)
      DO 75 I=1,LC4
      DO 80 K=1,LR4
      S=T1(I,K)
      TX=0.0
      IF(S.LE.0.)GO TO 85
      CALL SPLINE(S,XF,YF,N,2,TX,IL,VL,IU,VU,F,U)
   85 Y1(I,K)=TX
   80 CONTINUE
   75 CONTINUE
      RETURN
      END
      SUBROUTINE DELTA1(NUMM)
      COMMON/VALUES/R1(35,67),T1(35,67),RR(35,67),Y1(35,67),
     1X1(35,67),D1(35,67),E(35,67),E1(35,67),TBB(35,67),R4(35,67),
     1LC,LR,LC2,LR2,LR3,LC3,LR4,LC4,PLT01,PLOT2,PLOT3,PLOT4,PLOT5
      DATA ZERO/'0000'/
      DATA EYE/'    '/
      DATA STAR/'1111'/
      IF(NUMM.EQ.1)GO TO 50

      DO 10 I=1,LC2
      DO 20 J=1,LR2
      E(I,J)=0
      IF(D1(I,J).GT.0.AND.T1(I,J).GT.8.)E(I,J)=+1
      IF(D1(I,J).EQ.0.OR.T1(I,J).LT.8.)E(I,J)=0
      IF(D1(I,J).LT.0.AND.T1(I,J).GT.8.)E(I,J)=-1
   20 CONTINUE
   10 CONTINUE
      DO 30 N=1,LC2
      DO 40 L=1,LR2
```

```
      IF(E(N,L).EQ.+1)  E1(N,L)=ZERO
      IF(E(N,L).EQ.0)   E1(N,L)=EYE
      IF(E(N,L).EQ.-1)  E1(N,L)=STAR
   40 CONTINUE
   30 CONTINUE
      CALL WRITT(5)
      RETURN
   50 DO 60 I=1,LC4
      DO 70 J=1,LR4
      E(I,J)=0.
      IF(D1(I,J).GT.0.AND.T1(I,J).GT.8.)  E(I,J)=+1
      IF(D1(I,J).EQ.0.OR.T1(I,J).LT.8.)  E(I,J)=0
      IF(D1(I,J).LT.0.AND.T1(I,J).GT.8.)  E(I,J)=-1
   70 CONTINUE
   60 CONTINUE
      DO 80 N=1,LC4
      DO 90 L=1,LR4
      IF( E(N,L).EQ.+1)  E1(N,L)=ZERO
      IF( E(N,L).EQ.0)   E1(N,L)=EYE
      IF( E(N,L).EQ.-1)  E1(N,L)=STAR
   90 CONTINUE
   80 CONTINUE
      CALL WRITT(9)
      RETURN
      END
      SUBROUTINE CONTOU(TTT,LRW,LCW,NUMPLT,J)
      COMMON/OPTION/IMIN,IMAX,JMIN,JMAX,FINTER,NUMV,CONTUR(50)
      LOGICAL*1 FMT(5)/'F4.0)'/
      DIMENSION TTT(35,67)
      DATA NUMPT/0/
      NUMPT=NUMPT+1
      FINTER=0.0
      IMIN=1
      IMAX=LCW
      JMIN=1
      JMAX=LRW
      CALL SETGRD(1.,1.,15.,15.,4)
      CALL SETSIZ(1.)
      GO TO (10,20,30,40,40),J
   10 NUMV=11
      CONTUR(1)=190.
      CONTUR(2)=200.
      CONTUR(3)=210.
      CONTUR(4)=220.
      CONTUR(5)=230.
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
      CONTUR(6)=240.
      CONTUR(7)=250.
      CONTUR(8)=260.
      CONTUR(9)=270.
      CONTUR(10)=280.
      CONTUR(11)=290.
      CALL HORLIN('TEMPERATURE BRIGHTNESS',22,7.5,15.5,0,0)
      GO TO 50
   20 NUMV=13
      CONTUR(1)=200.
      CONTUR(2)=250.
      CONTUR(3)=300.
      CONTUR(4)=350.
      CONTUR(5)=400.
      CONTUR(6)=450.
      CONTUR(7)=500.
      CONTUR(8)=550.
      CONTUR(9)=600.
      CONTUR(10)=650.
      CONTUR(11)=700.
      CONTUR(12)=750.
      CONTUR(13)=800.
      CALL HORLIN('REFLECTANCES',12,7.5,15.5,0,0)
      GO TO 50
   30 NUMV=16
      CONTUR(1)=5.
      CONTUR(2)=10.
      CONTUR(3)=15.
      CONTUR(4)=20.
      CONTUR(5)=25.
      CONTUR(6)=30.
      CONTUR(7)=35.
      CONTUR(8)=40.
      CONTUR(9)=45.
      CONTUR(10)=50.
      CONTUR(11)=55.
      CONTUR(12)=60.
      CONTUR(13)=65.
      CONTUR(14)=70.
      CONTUR(15)=75.
      CONTUR(16)=80.
      CALL HORLIN('CLOUD OPTICAL THICKNESS',23,7.5,15.5,0,0)
      GO TO 50
   40 NUMV=10
      CONTUR(1)=-50.
```

```
      CONTUR(2)=-40.
      CONTUR(3)=-30.
      CONTUR(4)=-20.
      CONTUR(5)=-10
      CONTUR(6)=10.
      CONTUR(7)=20.
      CONTUR(8)=30.
      CONTUR(9)=40.
      CONTUR(10)=50.
      IF(J.EQ.5)GO TO 45
      CALL HORLIN('DELTA RATIO (+LIQUID-ICE)',25,7.5,15.5,-2,0)
      GO TO 50
   45 CALL HORLIN('DELTA RATIO (+LIQUID-ICE) WITH H2O VAPOR',40,7.5,
     115.5,-3.0)
   50 CONTINUE
      CALL SETSIZ(.50)
      CALL CONFRM(FMT(1))
      CALL INDEX(1)
      CALL LABELI(2)
      CALL OGRID(FLOAT(IMIN),FLOAT(IMAX),1,FMT,0,FLOAT(JMIN),
     1FLOAT(JMAX),1,FMT,0,0)
      CALL ZDUMP(TTT,35,67)
      CALL UNDEF(0.)
      CALL FILL(TTT,35,67,1)
      CALL TRACER(TTT,35,67)
      CALL FRMADV
      IF(NUMPLT.EQ.NUMPT)GO TO 60
      RETURN
   60 CALL SETBLK(999)
      CALL ENDPLT
      RETURN

      END
      SUBROUTINE WRITT(J)
      COMMON/VALUES/R1(35,67),T1(35,67),RR(35,67),Y1(35,67),
     1X1(35,67),D1(35,67),E(35,67),E1(35,67),TBB(35,67),R4(35,67),
     1LC,LR,LC2,LR2,LR3,LC3,LR4,LC4,PLOT1,PLOT2,PLOT3,PLOT4,PLOT5
      GO TO (10,20,30,40,50,60,70,80,90),J
   10 WRITE(9,1000)
      DO 15 K=1,LR
      WRITE(9,1100)(RR(I,K),I=1,LC)
   15 CONTINUE
      RETURN
   20 WRITE(9,1200)
      DO 25 K=1,LR
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
      WRITE(9,1100)(T1(I,K),I=1,LC)
   25 CONTINUE
      RETURN
   30 WRITE(9,1300)
      DO 35 K=1,LR2
      WRITE(9,1100)(RR(I,K),I=1,LC2)
   35 CONTINUE
      RETURN
   40 WRITE(9,1400)
      DO 45 K=1,LR2
      WRITE(9,1100)(RR(I,K),I=1,LC2)
   45 CONTINUE
      RETURN
   50 WRITE(9,1500)
      DO 55 K=1,LR2
      WRITE(9,1600)(E1(I,K),I=1,LC2)
   55 CONTINUE
      STOP
   60 WRITE(9,1700)
      DO 65 K=1,LR3
      WRITE(9,1100)(TBB(I,K),I=1,LC3)
   65 CONTINUE
      RETURN
   70 WRITE(9,1900)
      DO 75 K=1,LR4
      WRITE(9,1100)( RR(I,K),I=1,LC4)
   75 CONTINUE
      RETURN
   80 WRITE(9,2000)
      DO 85 K=1,LR4
      WRITE(9,1100)( RR(I,K),I=1,LC4)
   85 CONTINUE
      RETURN
   90 WRITE(9,1500)
      DO 95 K=1,LR4
      WRITE(9,1600)( E1(I,K),I=1,LC4)
   95 CONTINUE
      STOP
      RETURN
 1000 FORMAT(1H1,60X,'REFLECTANCES')
 1100 FORMAT(1H0,33F4.0)
 1200 FORMAT(1H1,54X,'CLOUD OPTICAL THICKNESS')
 1300 FORMAT(1H1,49X,'RELECTANCES RATIO R(1.61)/R(0.74)')
 1400 FORMAT(1H1,50X,'DELTA RATIO (+LIQUID-ICE)')
 1500 FORMAT(1H1,50X,'DELTA RATIO SYMBOL MAP')
```

```
 1600 FORMAT(1H0,33A4)
 1700 FORMAT(1H1,50X,'TEMPERATURE BRIGHTNESS')
 1900 FORMAT(1H1,48X,'REFLECTANCES RATIO R(1.61)/R(1.14)')
 2000 FORMAT(1H1,45X,'DELTA RATIO (+LIQUID-ICE) WITH H2O VAPOR')
      END
      SUBROUTINE SPLINE(S,X,Y,N,IN,T,IL,VL,IU,VU,E,U)
C     X,Y ARRAY OF IND. & DEPEN. VAR.,S THE ARGUMENT TO BE INTERPOLATED
C     T THE INTERPOLATED VALUE,N THE DIMENSION OF (X,Y)
C     IN=1 DETERMINES SPLINE FUNC.IN=2 INTERPOLATES SPLINE FNC
C     IL,IU=1 PARABOLIC RUNOUT CONDITIONS AT LOWER & UPPER BOUND.
C     IL,IU=2 1ST DERIVATIVE  (VL,VU) AT LOWER OR UPPER BOUND RESP.
C     IL,IU=3 2ND DERIVATIVE  (VL,VU) AT LOWER OR UPPER BOUND RESP.
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(1), Y(1), E(1),U(1)
      GO TO (10,40),IN
   10 CONTINUE
      N1=N-1
      B1=X(2)-X(1)
      C1=(Y(2)-Y(1))/B1
      GO TO (12,14,16),IL
   12 E(1)=1.0
      U(1)=0.0
      GO TO 18
   14 E(1)=-.5
      U(1)=(C1-VL)/2/B1
      GO TO 18
   16 E(1)=0
      U(1)=VL/12
   18 CONTINUE
      DO 20 J=2,N1
      B2=X(J+1)-X(J)
      C2=(Y(J+1)-Y(J))/B2
      B=X(J+1)-X(J-1)
      D=(C2-C1)/B
      C=B1/B
      B1=B2
      C1=C2
      P=C*E(J-1)+2.0
      E(J)=(C-1.0)/P
   20 U(J)=(D-C*U(J-1))/P
      GO TO (22,24,26),IU
   22 CONTINUE
      E(N)=U(N1)/(1.0-E(N1))
      GO TO 28
   24 C2=VU-(Y(N)-Y(N1))/(X(N)-X(N1))
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
      E(N)=(C2/(X(N)-X(N1))-U(N1))/(2.0D0+E(N1))
      GO TO 28
   26 E(N)=VU/12.0
   28 CONTINUE
      DO 30 KK=1,N1
      K=N-KK
      E(K)=E(K)*E(K+1)+U(K)
C     TO OBTAIN THE DERIVATIVES AT THE KNOTS REMOVE C FROM THE
C     FOLLOWING COMMENTS CARDS.THEN THE ARRAY U WILL CONTAIN
C     THE DERIVATIVES OF THE SPLINE FUNCTION AT THE KNOTS
      B2=X(K+1)-X(K)
      U(K)=(Y(K+1)-Y(K))/B2-B2*(E(K)*2+E(K+1))
   30 CONTINUE
      B2=X(N)-X(N1)
      U(N)=(E(N1)+2.0D0*E(N))*B2+(Y(N)-Y(N1))/B2
      RETURN
   40 IF (X(1).GT.X(N)) GO TO 50
      IDIR=0
      MLB=0
      MUB=N
      GO TO 60
   50 IDIR=1
      MLB=N
      MUB=0
   60 IF (S.GE.X(MUB+IDIR)) GO TO 100
      IF (S.LE.X(MLB+1-IDIR)) GO TO 110
      ML=MLB
      MU=MUB
      GO TO 80
   70 IF (IABS(MU-ML).LE.1) GO TO 120
   80 MAV=(ML+MU)/2
      IF (S.LT.X(MAV)) GO TO 90
      ML=MAV
      GO TO 70
   90 MU=MAV
      GO TO 70
  100 MU=MUB+2*IDIR
      GO TO 130
  110 MU=MLB+2*(1-IDIR)
      GO TO 130
  120 MU=MU+IDIR
  130 T=(E(MU-1)*((X(MU)-S)**3)+E(MU)*((S-X(MU-1))**3)+
     1   (Y(MU-1)-E(MU-1)*((X(MU)-X(MU-1))**2))*(X(MU)-S)+
     2   (Y(MU)-E(MU)*((X(MU)-X(MU-1))**2))*(S-X(MU-1)))/
     3   (X(MU)-X(MU-1))
```

RETURN
END

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

APPENDIX I - QUICK3 PROGRAM LISTINGS, INPUT, AND OUTPUT JCL

```
C0000100  // EXEC LINKGO,REGION.GO=475K
C0000150  //LINK.SYSLIB DD DSN=SYS2.WCLFTEST,DISP=SHR
C0000190  //LINK.OBJECT DD *
C0000195    INCLUDE LOADLIB(ZBMJMR92)
C0000200  //*     UNIT 6 IS THE DEBUG AND GENERAL MESSAGE UNIT ON PRINTER
C0000300  //*       UNIT 9 CONTAINS THE LISTING OF THE TAPE CONTENTS
C0000400  //*         TP192 IS THE INPUT DATA TAPE
C0000500  //GO.FT06F001 DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=7265),
C0000600  //  SPACE=(CYL,(7,3))
C0000700  //GO.FT09F001 DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=7265),
C0000800  //  SPACE=(CYL,(15,5))
C0000900  //GO.TP192 DD UNIT=2400-9,DCB=(RECFM=U,BLKSIZE=32000),LABEL=(,BLP),
C0001000  //  VOL=SER=L0045
C0001100  //GO.PLOTTAPE DD DCB=(,DEN=1),LABEL=(,BLP,,OUT),UNIT=(7TRACK,,DEFER),
C0001200  //  VOL=SER=L0046
C0001300  //GO.SYSUDUMP DD SYSOUT=A
C0001400  //GO.DATAS DD *
```

*** END OF MEMBER *** 17 RECORDS PROCESSED.

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
C*************************************************************************
C   THIS PROGRAM WAS WRITTEN BY F. MILASI OF CSC FOR TASK 092,
C SKYLAB, UNDER CONTRACT NAS-5-11999 IN MAY, 1974. IT UTILIZES
C SUBROUTINES TEMBRH, REFLE1, CLUTHK, SPLINE, CONTOU.
C   USING THE READ PROGRAM ALREADY DEVELOPED FOR THE S192 DATA,
C THIS       DATA SURVEY PLOTTING PACKAGE, QUICKPL, SELECTS
C SUBSETS OF THE DATA FOR WHICH REFLECTANCE, RATIOS OF REFLECTANCE,
C OPTICAL THICKNESS, BRIGHTNESS TEMPERATURES AND OTHER SIGNIFICANT
C PARAMETERS ARE CALCULATED AND DISPLAYED THROUGH A CONTOUR PICTURE.
C USING THE WOLF PLOT PACKAGE. CONTOU WILL BE USED TO DISPLAY THE
C OUTPUT  OF ONE OF THE FOUR CHANNELS WHICH QUICKPL HAS ANALYZED.
C CONTOU CAN PROCESS ONLY ONE CHANNEL PER RUN BECAUSE OF CORE
C REQUIREMENT DIFFICULTIES AND,BECAUSE OF THIS,CONTOU WILL ONLY
C DISPLAY  A 200X200 AREA. A COMPLETE PICTURE WILL BE FORMED
C THROUGH THE ASSEMBLING OF OVERLAPPING 200X200 AREAS.
C*************************************************************************
C
C*************************************************************************
C  THE VALUES WILL BE PASSED TO THE OTHER SUBROUTINES THROUGH THE
C 'COMMON' NAMED'VALUE'. LR3, LC3 TELL HOW LARGE THE ARRAY TBB
C IS.ALSO PLTTEM, PLTREF, PLTCLD ARE LOGICAL VARIABLES ARE
C EITHER .TRUE. OR .FALSE. DEPENDING ON WHAT CONTOUR THE USER
C HAS REQUESTED. THE SOLAR ZENITH ANGLE THAT WILL BE USED IN THE
C CALCULATION OF  REFLECTANCE WILL BE STORED IN 'ANG'.
C*************************************************************************
C
      COMMON/VALUE/LR3,LC3,TBB(200,200),PLTTEM,PLTREF,PLTCLD,ANG
      DIMENSION SCAN(1098),ISCAN(1098),NVAL(3),PIXEL(1036)
      INTEGER*2 ISCAN2(2194),DEG,AMIN,OSEC
C*************************************************************************
C  NAMELIST 'INPUT' WILL BE USED TO INPUT THE DESIRED CONTOUR, CHANNEL
C NUMBER, NUMBER OF PIXEL, STARTING NUMBER LINE, AND STARTING PIXEL.
C      NAMELIST NCHAN1=CHANNEL NUMBER
C               NUMPIT=NUMBER OF PIXEL POINTS TO INCREMENT BY
C               IPIXEL=STARTING PIXEL
C               ISTLIN=STARTING LINE
C               PLTTEM=TEMPERATURE BRIGHTNESS PLOT
C               PLTREF=REFLECTANCE PLOT
C               PLTCLD=CLOUD OPTICAL THICKNESS PLOT
C  THE ABOVE NAMELIST HAS   DEFAULT OPTIONS:        NCAHN1=7,
C NUMPIT=1, IPIXEL=2, ISTLIN=1, PLTTEM=.TRUE., PLTREF=.FALSE.,
C PLTCLD=.FALSE..
C*************************************************************************
C
      NAMELIST/INPUT/ NCHAN1, NUMPIT,NUMLIN,IPIXEL,ISTLIN,
```

```
     1PLTTEM,PLTREF,PLTCLD
      LOGICAL*4 PLTTEM,PLTREF,PLTCLD
      DATA KCOUNT/0/
      DATA ISTLIN/1/
      DATA IPIXEL/2/
      DATA NUMPIT/1/
      DATA NUMLIN/1/
      DATA NVAL/0,0,0/
      DATA RADI/.017453/
      EQUIVALENCE (NVAL(1),NCOND), (NVAL(2),NHOUS),(NVAL(3),NSCAN)
      EQUIVALENCE (SCAN(1),ISCAN(1),ISCAN2(1))
      EQUIVALENCE (ISCAN2(50),DEG),(ISCAN2(51),AMIN),
     1(ISCAN2(52),QSEC)

      EQUIVALENCE (PIXEL(1),SCAN(62))
      DATA NCOUNT/0/
      DATA NCOU1/0/
      DATA LASTRW/0/
      DATA LASTCL/0/
      PLTTEM=.FALSE.
      PLTREF=.FALSE.
      PLTCLD=.FALSE.
C*************************************************************************
C    READING IN INPUT CARDS 'NAMELIST'
C*************************************************************************
      READ(5, INPUT)
C*************************************************************************
C  CHECKING WHAT CHANNEL HAS BEEN REQUESTED AGAINST CHANNELS 15, 16
C THESE CHANNELS ARE NOT ACTIVE CHANNELS.
C*************************************************************************
      IF(NCHAN1.EQ.15.OR.NCHAN1.EQ.16)GO TO 90
      KT=ISTLIN
      IF(KT.LT.NUMLIN)GO TO 5
      IN=KT/NUMLIN
      KT=KT-IN*NUMLIN-1
    5 CONTINUE
      LR3=0
      LC3=0
   10 CONTINUE
      NCOUNT=NCOUNT+1
C*************************************************************************
C    CALLING S192 READ PROGRAM WRITTEN BY M. MACK OF CSC FOR TASK
C 092, SKYLAB, UNDER CONTRACT NAS-5-11999.
C*************************************************************************
      CALL RD192(SCAN,NVAL)
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
      IF(NCOND.EQ.1)GO TO 70
      IF(NCOND.EQ.2)GO TO 80
      IF(ISCAN2(4).LT.ISTLIN)GO TO 10
      IF(ISCAN2(1).EQ.NCHAN1)GO TO 20
      GO TO 10
   20 CONTINUE
C*************************************************************************
C   READING AND SETTING UP THE DISERED NUMBER OF LINES FOR ARRAY
C TBB   (200 LINES MAX).
C*************************************************************************
      NCOU1=NCOU1+1
      LINEN1=ISCAN2(4)
      LT=LINEN1-KT
      NROWM=MOD(LT,NUMLIN)
      IF(NUMLIN.NE.1.AND.NROWM.NE.1)GO TO 10
      NROW1=((LINEN1-ISTLIN)/NUMLIN)+1
      IF(LASTRW.GT.200)GO TO 50
      LASTRW=NROW1
      NCOL1=0
C*************************************************************************
C  READING IN THE DESIRED NUMBER OF PIXEL POINTS INTO ARRAY TBB..
C .....................(200 PIXEL POINTS MAX).................
C*************************************************************************
      DO 25 N=1PIXEL,1036,NUMPIT
      NCOL1=NCOL1+1
      LASTCL=NCOL1
      IF(NCOL1.GT.200)GO TO 25
      TBB(NCOL1,NROW1)=PIXEL(N)
   25 CONTINUE
C*************************************************************************
C  CHECKING LR3 AND LC3 TO SEE IF GREATER THAN 200, IF SO SETTING
C THE ARRAY TBB TO 200X200.
C*************************************************************************
      LR3=LASTRW
      IF(LASTRW.GT.200)LR3=200
      LC3=LASTCL
      IF(LASTCL.GT.200)LC3=200
C*************************************************************************
C   CALCULATION OF THE SOLOR ZENITH ANGLE
C*************************************************************************
      KCOUNT=KCOUNT+1
      IF(KCOUNT.GT.1)GO TO 10
      ANGG=DEG+AMIN/60.+0SEC/240.
      ANG=ANGG*RADI
      GO TO 10
```

```
   50 CONTINUE
      WRITE(6,3000)NCOUNT,NCOU1,LR3,LC3
      WRITE(6, INPUT)
C*****************************************************************
C    CHECKING LOGICAL VARIABLE TO SEE IF THIS CONTOUR PLOT WAS
C REQUESTED. IF .TRUE. SUBROUTINE TEMBRH WILL BE CALLED, IF
C .FALSE. SUBROUTINE REFLE1 WILL BE CALLED.
C*****************************************************************
      IF(PLTTEM)CALL TEMBRH(NCHAN1)
      IF(PLTREF)CALL REFLE1(NCHAN1)
      IF(PLTCLD)CALL REFLE1(NCHAN1)
      STOP

   70 WRITE(6,1000)
      STOP
   80 WRITE(6,2000)NCOUNT,NCOU1,LR3,LC3
      WRITE(6, INPUT)
      IF(PLTTEM)CALL TEMBRH(NCHAN1)
      IF(PLTREF)CALL REFLE1(NCHAN1)
      STOP
   90 WRITE(6,4000)NCHAN1
      STOP
 1000 FORMAT(1HO,10X,'TAPE ERROR')
 2000 FORMAT(1HO,10X,'END OF FILE',5X,'NCOUNT=',I6,2X,'NCOU1=',I5,2X,
     1'LR3=',I3,2X,'LC3=',I3)
 3000 FORMAT(1HO,10X,'CALL BEFORE END OF FILE' 'NCOUNT=',I6,
     1'NCOU1=',I4,'LR3=',I4,'LC3=',I4)
 4000 FORMAT(1HO,10X,'PLEASE CHECK CHANNEL THAT YOU REQUESTED,
     1CHANNELS 15,16    ARE NOT ACTIVE',5X,'NCHAN1=',I3)
     2'NCHANE2='I3)
      END
      SUBROUTINE TEMBRH(NCHAN1)
      COMMON/VALUE/LR3,LC3,TBB(200,200),PLTTEM,PLTREF,PLTCLD
      LOGICAL*4 PLTTEM,PLTREF,PLTCLD
      DIMENSION CHANNL(21)
      DATA CONE/3.7413E+04/
      DATA CTWO/1.4388E+04/
      DATA PI/3.1416/
      DATA CHANNL/.540,.540,.585,.585,.645,.645,.720,.720,
     1.830,.830,1.650,1.650,2.225,2.225,.000,.000,12.500,
     2.485,1.005,1.140,11.350/
      DO 10 I=1,21
      WAVET=CHANNL(I)
      IF(NCHAN3.EQ.I)GO TO 15
   10 CONTINUE
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
   15 TTT=PI*WAVET**5
      DO 20 I=1,LC3
      DO 30 J=1,LR3

      IF(TBB(I,J).LE.0.)GO TO 25
      TAB=1.+(CONE/(TTT*TBB(I,J)))
      TAA=WAVET*ALOG(TAB)
      TBB(I,J)=CTWO/TAA
      GO TO 30
   25 TBB(I,J)=0.0
   30 CONTINUE
   20 CONTINUE
      CALL CONTOU
      RETURN
      END
      SUBROUTINE REFLE1(NCHAN1)
C
C  THIS SUBROUTINE WAS WRITTEN BY F. MILASI OF CSC FOR TASK 092,
C SKYLAB, UNDER CONTRACT NAS-5-11999 IN APRIL, 1974. SUBROUTINE
C REFLE1 TAKES MATRIX 'A', WHICH WAS PASSED FROM THE MAIN
C PROGRAM AND  WAS FORMED FROM NCHAN1.
C  THIS WILL BE TRANSFORMED INTO REFLECTANCES BY THE MULTIPLICATION
C OF EACH ELEMENT BY THE FACTOR (PI/FO*COS(ANG)).    THE
C RESULTING MATRIX 'R1' WILL BE MULTIPLIED BY 1000. (TO ALLOW
C FIXED POINT FORMAT) AND WILL BE PRINTED OUT BY SUBROUTINE WRITT.
C**********************************************************************
C**********************************************************************
      COMMON/VALUE/LR3,LC3,TBB(200,200),PLTTEM,PLTREF,PLTCLD,ANG
C
C**********************************************************************
C  ARRAY FF IS THE ARRAY OF IRRADIANCE DATA ASSOCIATED WITH THE
C AVERAGE WAVELENGTH FOR EACH ACTIVE CHANNED. THESE VALUES WERE
C DERIVED THRU GAUSSIAN 'AVERAGING' OF DR. THEKAKARA IRRADIANCE
C DATA USING  CNTHGS92.
C       NOTE: THE FF IRRADIANCE DATA GIVEN FOR THE 15TH AND 19TH
C ACTIVE CHANNELS ARE MERELY DUMMY VALUES. SINCE IRRADIANCE
C DATA ASSOCIATED WITH THE WAVELENGTHS REPUSENTED BY THOSE
C CHANNELS ARE EXTREMELY SMALL AND INSIGNIFICANT, NO
C CALCULAT USING THEM WOULD BE SCIENTIFICALLY MEANINGFUL,
C (ACCORDING TO DR. CURRAN OF GSFC). SO, AS THESE CHANNELS
C WOULD NEVER BE USED. DERVATION OF ACCURATE IRRADIANCE VALUES
C SEEMED POINTLESS.
C**********************************************************************
      LOGICAL*4 PLTTEM,PLTREF,PLTCLD
      REAL*8 FF(22),FO
```

```
      DATA FF/0.1784488D0,0.1784488D0,0.1704564D0,0.1704564D0,
     10.1526697D0,0.1526697D0,0.1310974D0,0.1310974D0,0.1034361D0,
     10.1034361D0,0.0220749D0,0.0220749D0,0.0081098D0,0.0081098D0,
     10.0000000D0,0.0000000D0,0.0000000D0,0.1999891D0,0.0762900D0,
     10.0540257D0,0.0000000D0,0.1723639D0/
      DATA PI/3.1416/
      DO 5 M=1,22
      FO=FF(M)
      IF(M.EQ.NCHAN1)GO TO 7
    5 CONTINUE
      WRITE(6,1000)NCHAN1
      STOP
    7 CONTINUE
      CA=COS(ANG)
      F=FO*CA
      FAC=PI/F
      DO 10 I=1,LC3
      DO 20 J=1,LR3
      IF(TBB(I,J).LE.0)GO TO 15
      TBB(I,J)=TBB(I,J)*FAC
      GO TO 20
   15 TBB(I,J)=0.0
   20 CONTINUE
   10 CONTINUE
      IF(PLTCLD) CALL CLUTHK 0
      DO 30 L=1,LC3
      DO 40 N=1,LR3
      TBB(L,N)=TBB(L,N)*1000.
   40 CONTINUE
   30 CONTINUE
      IF(PLTREF)CALL CONTOU
      STOP
 1000 FORMAT(1HO,10X,'NCHAN1 DOES NOT AGREE WITH FF ARRAY',
     15X,'NCHAN1=',I3)
      END
      SUBROUTINE CLUTHK
      COMMON/VALUE/LR3,LC3,TBB(200,200),PLTTEM,PLTREF,PLTCLD
      DIMENSION X(10),Y(10),F(10),U(10)
      LOGICAL*4 PLTTEM,PLTREF,PLTCLD
      REAL*8 S,X,Y,TX,VL,VU,F,U
      DATA X/0.,0.038,0.092,0.248,0.422,0.607,0.759,0.848,
     10.892,0.920/
      DATA Y/0.,1.,2.,5.,10.,20.,50.,100.,200.,500./
    5 N=10
      IL=3
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
      IU=3
      VL=0.
      VU=0.
      CALL SPLINE(S,X,Y,N,1,TX,IL,VL,IU,VU,F,U)
      DO 10 L=1,LC3
      DO 20 M=1,LR3

      S=TBB(L,M)
      TX=0.0
      IF(S.LE.0.)GO TO 15
      CALL SPLINE(S,X,Y,N,2,TX,IL,VL,IU,VU,F,U)
   15 TBB(L,M)=TX
   20 CONTINUE
   10 CONTINUE
      CALL CONTOU
      RETURN
      END
      SUBROUTINE CONTOU
      COMMON/VALUE/LR3,LC3,TBB(200,200),PLTTEM,PLTREF,PLTCLD

      COMMON/OPTION/IMIN,IMAX,JMIN,JMAX,FINTER,NUMV,CONTUR(50)
      COMMON/BUF$/BUFFER(4600)
      LOGICAL*1 FMT(5)/'F4.0)'/
      LOGICAL*4 PLTTEM,PLTREF,PLTCLD
      FINTER=0.0
      IMIN=1
      IMAX=LC3
      JMIN=1
      JMAX=LR3
      CALL NOSTAE
      CALL PLOTST(1001,4)
      CALL SETSIZ(1.)
      IF(PLTTEM)GO TO 10
      IF(PLTREF)GO TO 20
      IF(PLTCLD)GO TO 30
      STOP
   10 NUMV=26
      CONTUR(1)=278.
      CONTUR(2)=279.
      CONTUR(3)=280.
      CONTUR(4)=281.
      CONTUR(5)=282.
      CONTUR(6)=283.
      CONTUR(7)=284.
      CONTUR(8)=285.
```

```
      CONTUR(9)=286.
      CONTUR(10)=287.
      CONTUR(11)=288.
      CONTUR(12)=289.
      CONTUR(13)=290.
      CONTUR(14)=291.
      CONTUR(15)=292.
      CONTUR(16)=293.
      CONTUR(17)=294.
      CONTUR(18)=295.
      CONTUR(19)=296.
      CONTUR(20)=297.
      CONTUR(21)=298.
      CONTUR(22)=299.
      CONTUR(23)=300.
      CONTUR(24)=301.
      CONTUR(25)=302.
      CONTUR(26)=303.
      CALL HORLIN('TEMPERATURE BRIGHTNESS',22,10.8,22.0,-3,0)
      GO TO 40
   20 NUMV=13
      CONTUR(1)=200.
      CONTUR(2)=250.
      CONTUR(3)=300.
      CONTUR(4)=350.
      CONTUR(5)=400.
      CONTUR(6)=450.
      CONTUR(7)=500.
      CONTUR(8)=550.
      CONTUR(9)=600.
      CONTUR(10)=650.
      CONTUR(11)=700.
      CONTUR(12)=750.
      CONTUR(13)=800.
      CALL HORLIN('REFLECTANCES',12,12.5,25.5,-2,0)
      GO TO 40
   30 NUMV=16
      CONTUR(1)=5.
      CONTUR(2)=10.
      CONTUR(3)=15.
      CONTUR(4)=20.
      CONTUR(5)=25.
      CONTUR(6)=30.
      CONTUR(7)=35.
      CONTUR(8)=40.
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
      CONTUR(9)=45.
      CONTUR(10)=50.
      CONTUR(11)=55.
      CONTUR(12)=60.
      CONTUR(13)=65.
      CONTUR(14)=70.
      CONTUR(15)=75.
      CONTUR(16)=80.
      CALL HORLIN('CLOUD OPTICAL THICKNESS',23,12.5,25.5,-5,0)
   40 CALL SETGRD(1.,1.,21.6,21.6,4)
      CALL SETSIZ(.60)
      CALL CONFRM(FMT(1))
      CALL INDEX(1)
      CALL LABELI(2)
      CALL OGRID(FLOAT(IMIN),FLOAT(IMAX),1,FMT,0,FLOAT(JMIN),
     1FLOAT(JMAX),1,FMT,0,0)
      CALL UNDEF(0.0)
      CALL FILL(TBB,200,200,1)
      CALL TRACER(TBB,200,200)
      CALL FRMADV
      CALL SETBLK(999)
      CALL ENDPLT
      RETURN
      END
      SUBROUTINE SPLINE(S,X,Y,N,IN,T,IL,VL,IU,VU,E,U)
C     X,Y ARRAY OF IND. & DEPEN. VAR.,S THE ARGUMENT TO BE INTERPOLATED
C     T THE INTERPOLATED VALUE,N THE DIMENSION OF (X,Y)
C     IN=1 DETERMINES SPLINE FUNC,IN=2 INTERPOLATES SPLINE FNC
C     IL,IU=1 PARABOLIC RUNOUT CONDITIONS AT LOWER & UPPER BOUND.
C     IL,IU=2 1ST DERIVATIVE  (VL,VU) AT LOWER OR UPPER BOUND RESP.
C     IL,IU=3 2ND DERIVATIVE  (VL,VU) AT LOWER OR UPPER BOUND RESP.
      IMPLICIT REAL*8(A-H,O-Z)
      DIMENSION X(1), Y(1), E(1),U(1)
      GO TO (10,40),IN
   10 CONTINUE
      N1=N-1
      B1=X(2)-X(1)
      C1=(Y(2)-Y(1))/B1
      GO TO (12,14,16),IL
   12 E(1)=1.0
      U(1)=0.0
      GO TO 18
   14 E(1)=-.5
      U(1)=(C1-VL)/2/B1
      GO TO 18
```

```
   16 E(1)=0
      U(1)=VL/12
   18 CONTINUE
      DO 20 J=2,N1
      B2=X(J+1)-X(J)
      C2=(Y(J+1)-Y(J))/B2
      B=X(J+1)-X(J-1)
      D=(C2-C1)/B
      C=B1/B
      B1=B2
      C1=C2
      P=C*E(J-1)+2.0
      E(J)=(C-1.0)/P
   20 U(J)=(D-C*U(J-1))/P
      GO TO (22,24,26),IU
   22 CONTINUE
      E(N)=U(N1)/(1.0-E(N1))
      GO TO 28
   24 C2=VU-(Y(N)-Y(N1))/(X(N)-X(N1))
      E(N)=(C2/(X(N)-X(N1))-U(N1))/(2.0D0+E(N1))
      GO TO 28
   26 E(N)=VU/12.0
   28 CONTINUE
      DO 30 KK=1,N1
      K=N-KK
      E(K)=E(K)*E(K+1)+U(K)
C     TO OBTAIN THE DERIVATIVES AT THE KNOTS REMOVE C FROM THE
C     FOLLOWING COMMENTS CARDS.THEN THE ARRAY U WILL CONTAIN
C     THE DERIVATIVES OF THE SPLINE FUNCTION AT THE KNOTS
      B2=X(K+1)-X(K)
      U(K)=(Y(K+1)-Y(K))/B2-B2*(E(K)*2+E(K+1))
   30 CONTINUE
      B2=X(N)-X(N1)
      U(N)=(E(N1)+2.0D0*E(N))*B2+(Y(N)-Y(N1))/B2
      RETURN
   40 IF (X(1).GT.X(N)) GO TO 50
      IDIR=0
      MLB=0
      MUB=N
      GO TO 60
   50 IDIR=1
      MLB=N
      MUB=0
   60 IF (S.GE.X(MUB+IDIR)) GO TO 100
      IF (S.LE.X(MLB+1-IDIR)) GO TO 110
```

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

```
      ML=MLB
      MU=MUB
      GO TO 80
   70 IF (IABS(MU-ML).LE.1) GO TO 120
   80 MAV=(ML+MU)/2
      IF (S.LT.X(MAV)) GO TO 90
      ML=MAV
      GO TO 70
   90 MU=MAV
      GO TO 70
  100 MU=MUB+2*IDIR
      GO TO 130
  110 MU=MLB+2*(1-IDIR)
      GO TO 130
  120 MU=MU+IDIR
  130 T=(E(MU-1)*((X(MU)-S)**3)+E(MU)*((S-X(MU-1))**3)+
     1    (Y(MU-1)-E(MU-1)*((X(MU)-X(MU-1))**2))*(X(MU)-S)+
     2    (Y(MU)-E(MU)*((X(MU)-X(MU-1))**2))*(S-X(MU-1)))/
     3    (X(MU)-X(MU-1))
      RETURN
      END
```

APPENDIX J - JSC-SUPPLIED TAPE FORMATS FOR S-191 AND S-192 TAPES

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

5.3.2 S191 Processed Data Tape

This section describes the processed data tape.

S191 PROCESSED HEADER RECORD

BYTE NO.	CONTENT	DESCRIPTION
1-2	2520	RECORD LENGTH (BYTES)
3-8		DATE OF TAPE GENERATION
3-4	28	DAY
5-6	09	MONTH } EXAMPLE: 28 SEPT.72
7-8	72	YEAR
9-14		START DATE OF DATA
9-10	01	DAY
11-12	09	MONTH } EXAMPLE: 1 SEPT.72
13-14	72	YEAR
15-16	21	NO. OF MEASUREMENTS DESCRIBED
17-18	616	DATA BLOCK LENGTH (BYTES)
19-20	4	NO. OF DATA BLOCKS PER DATA RECORD
21-50		MEASUREMENT NUMBER 1
		SEGMENT 1 CHANNEL A2 RAD- WATTS/ CM^2/ MICRON/STER
21-22	5	START LOCATION (BYTE NO.)
23-24	20	NO. OF SAMPLES PER BLOCK
25-26	16	NO. OF BITS PER MEASUREMENT
27-34	P102-191	MEASUREMENT NUMBER
35-36	A_0 COEFFICIENT	INTERCEPT
37-38	E_0 EXPONENT	
39-40	A_1 COEFFICIENT	SLOPE
41-42	E_1 EXPONENT	
43-46	0^1	ΔT
47-50	0	Δt
51-52	45	
53-54	20	
55-56	16	MEASUREMENT NUMBER 2
57-64	P103-191	SEGMENT 1 CHANNEL A3 RAD. - WATTS/CM2/MICRON/ STER
73-76	0	
77-80	0	
81-82	85	
83-84	20	
85-86	16	MEASUREMENT NUMBER 3
87-94	P105-191	SEGMENT 1 CHANNEL A5 RAD.-WATTS/CM^2/MICRON/ STER
103-106	0	
107-110	0	

BYTE NO.	CONTENT	DESCRIPTION
111-112	125	
113-114	20	
115-116	16	MEASUREMENT NUMBER 4
117-124	P202-191	SEGMENT 2 CHANNEL A2 RAD - WATTS/CM^2/ MICRON/STER
133-136	0	
137-140	0	
141-142	165	
143-144	20	
145-146	16	MEASUREMENT NUMBER 5
147-154	P203-191	SEGMENT 2 CHANNEL A3 RAD - WATTS/CM^2/ MICRON/STER
163-166	0	
167-170	0	
171-172	205	
173-174	20	
175-176	16	MEASUREMENT NUMBER 6
177-184	P205-191	SEGMENT 2 CHANNEL A5 RAD - WATTS/CM^2/ MICRON/STER
193-196	0	
197-200	0	
201-202	245	
203-204	20	
205-206	16	MEASUREMENT NUMBER 7
207-214	P302-191	SEGMENT 3 CHANNEL A2 RAD - WATTS/CM^2/ MICRON/STER
223-226	0	
227-230	0	
231-232	285	
233-234	20	
235-236	16	MEASUREMENT NUMBER 8
237-244	P303-191	SEGMENT 3 CHANNEL A3 RAD - WATTS/CM^2/ MICRON/STER
253-256	0	
257-260	0	

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

PHO-TR543
Volume 1
Change 1
23 May 1973

S191 PROCESSED HEADER RECORD (CONTINUED)

BYTE NO.	CONTENT	DESCRIPTION	BYTE NO.	CONTENT	DESCRIPTION
261-262	325		411-412	525	
263-264	20		413-414	20	
265-266	16	MEASUREMENT NUMBER 9	415-416	16	MEASUREMENT NUMBER 14
267-274	P305-191	SEGMENT 3 CHANNEL A5 RAD. WATTS/CM^2/MICRON/STER	417-424	P601-191	SEGMENT 6 CHANNEL A1 RAD. WATTS/CM^2/ MICRON/STER
⋮			⋮		
283-286	0		433-436	0	
287-290	0		437-440	0	
291-292	365		441-442	565	
293-294	20		443-444	20	
295-296	16	MEASUREMENT NUMBER 10	445-446	16	MEASUREMENT NUMBER 15
297-304	P401-191	SEGMENT 4 CHANNEL A1 RAD - WATTS/CM^2/MICRON/STER	447-454	P606-191	SEGMENT 6 CHANNEL A6 RAD. - WATTS/CM^2/ MICRON/STER
⋮			⋮		
313-316	0		463-466	0	
317-320	0		467-470	0	
321-322	405		471-472	605	
323-324	20		473-474	1	
325-326	16	MEASUREMENT NUMBER 11	475-476	16	MEASUREMENT NUMBER 16*
327-334	P406-191	SEGMENT 4 CHANNEL A6 RAD - WATTS/CM^2/MICRON/STER	477-484	P026-SKY	GEODETIC LATITUDE - DEG
⋮			⋮		
343-346	0		493-496	0	
347-350	0		497-500	0	
351-352	445		501-502	607	
353-354	20		503-504	1	
355-356	16	MEASUREMENT NUMBER 12	505-506	16	MEASUREMENT NUMBER 17*
357-364	P501-191	SEGMENT 5 CHANNEL A1 RAD - WATTS/CM^2/MICRON/STER	507-514	P023-SKY	LONGITUDE - DEG
⋮			⋮		
373-376	0		523-526	0	
377-380	0		527-530	0	
381-382	485		531-532	609	
383-384	20		533-534	1	
385-386	16	MEASUREMENT NUMBER 13	535-536	16	MEASUREMENT NUMBER 18*
387-394	P506-191	SEGMENT 5 CHANNEL A6 RAD - WATTS/CM^2/MICRON/STER	537-544	P025-SKY	ALTITUDE - KILOMETERS
⋮			⋮		
403-406	0		553-556	0	
407-410	0		557-560	0	

*THESE MEASUREMENTS ARE INCLUDED SO AS TO MAKE THE PROCESSED TAPE FORMATS FOR S191 AND THE FIELD SPECTROMETER COMPATIBLE ON THE TAPE CONTAINING FIELD SPECTROMETER DATA THESE FIELDS SHALL CONTAIN FILL ZEROS.

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

S191 PROCESSED HEADER RECORD (CONTINUED)

BYTE NO.	CONTENT	DESCRIPTION
561-562	611	
563-564	1	
565-566	16	MEASUREMENT NUMBER 19*
567-574	P073-SKY	DRIFT ZLV YAW - DEG
⋮		
583-586	0	
587-590	0	
591-592	613	
593-594	1	
595-596	16	MEASUREMENT NUMBER 20*
597-604	P071-SKY	ROLL ZLV - DEG
⋮		
613-616	0	
617-620	0	
621-622	615	
623-624	1	
625-626	16	MEASUREMENT NUMBER 21*
627-634	P072-SKY	PITCH ZLV - DEG
⋮		
643-646	0	
647-650	0	
651-710		60 CHARACTERS OF ALPHANUMERIC IDENTIFICATION (ITITLE)
711-712		NO. OF ACTIVE WAVELENGTHS IN FILTER SEGMENT 1
713-714		NO. OF ACTIVE WAVELENGTHS IN FILTER SEGMENT 2
715-716		NO. OF ACTIVE WAVELENGTHS IN FILTER SEGMENT 3
717-718		NO. OF ACTIVE WAVELENGTHS IN FILTER SEGMENT 4
719-720		NO. OF ACTIVE WAVELENGTHS IN FILTER SEGMENT 5
721-722		NO. OF ACTIVE WAVELENGTHS IN FILTER SEGMENT 6
723-762		20 WAVELENGTHS (2 BYTES PER WAVELENGTH) FOR FILTER SEGMENT 1
763-802		20 WAVELENGTHS (2 BYTES PER WAVELENGTH) FOR FILTER SEGMENT 2
803-842		20 WAVELENGTHS (2 BYTES PER WAVELENGTH) FOR FILTER SEGMENT 3
843-882		20 WAVELENGTHS (2 BYTES PER WAVELENGTH) FOR FILTER SEGMENT 4
883-922		20 WAVELENGTHS (2 BYTES PER WAVELENGTH) FOR FILTER SEGMENT 5
923-962		20 WAVELENGTHS (2 BYTES PER WAVELENGTH) FOR FILTER SEGMENT 6
963-2520		FILL ZEROS

*THESE MEASUREMENTS ARE INCLUDED SO AS TO MAKE THE PROCESSED TAPE FORMATS FOR S191 AND THE FIELD SPECTROMETER COMPATIBLE. ON THE TAPE CONTAINING FIELD SPECTROMETER DATA THESE FIELDS SHALL CONTAIN FILL ZEROS.

S191 PROCESSED DATA RECORD

RECORD BYTE NO.	CONTENT
1-4	FRAME TIME
5-44	20 SAMPLES OF P102-191
45-84	20 SAMPLES OF P103-191
85-124	20 SAMPLES OF P105-191
125-164	20 SAMPLES OF P202-191
165-204	20 SAMPLES OF P203-191
205-244	20 SAMPLES OF P205-191
245-284	20 SAMPLES OF P302-191
285-324	20 SAMPLES OF P303-191
325-364	20 SAMPLES OF P305-191
365-404	20 SAMPLES OF P401-191
405-444	20 SAMPLES OF P406-191
445-484	20 SAMPLES OF P501-191
485-524	20 SAMPLES OF P506-191
525-564	20 SAMPLES OF P601-191
565-604	20 SAMPLES OF P606-191
605-606	1 SAMPLE OF P026-SKY
607-608	1 SAMPLE OF P023-SKY
609-610	1 SAMPLE OF P025-SKY
611-612	1 SAMPLE OF P073-SKY
613-614	1 SAMPLE OF P071-SKY
615-616	1 SAMPLE OF P072-SKY
617-1232	BLOCK 2
1233-1848	BLOCK 3
1849-2464	BLOCK 4
2465-2520	FILL ZEROS

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

S-192 CYBER 73 LINE STRAIGHTENED DATA TAPE HEADER RECORD

BYTES	CONTENTS	DESCRIPTION
1-32	PRODUCTION	COMPUTING SYSTEM PRODUCING (EXAMPLE) THIS TAPE (EBCDIC)
33-52		TAPE LIBRARY ID-EBCDIC
53-60	S-192	SENSOR ID - EBCDIC
61-63		DATE OF THIS TAPE GENERATION (EXAMPLE)
	21	DAY OF MONTH - BINARY
	09	MONTH NUMBER - BINARY
	72	YEAR - LAST 2 DIGITS - BINARY
64	1	TAPE SEQUENCE ID-BINARY (EXAMPLE)
65-66	0	MISSION NO. BINARY (EXAMPLE)
67-68	37	SITE - BINARY (EXAMPLE)
69	0	LINE - BINARY (EXAMPLE)
70	0	RUN - BINARY (EXAMPLE)
71-72	543	ORBIT - BINARY (EXAMPLE)
73-80		TIME OF FIRST SCAN IN THIS JOB (EXAMPLE)
73-74	183	TENTHS OF MILLISEC - BINARY
75	42	SECONDS - BINARY
76	15	MINUTES - BINARY
77	11	HOURS - BINARY
78	21	DAY OF MONTH - BINARY
79	09	MONTH NUMBER - BINARY
80	72	YEAR - LAST 2 DIGITS - BINARY
81-88	CHANNELS 1 THRU 22	CHANNELS ACTIVE IN THIS JOB (EXAMPLE) UP TO 64 CHANNELS, 1 BIT/CH STARTING LEFT TO RIGHT (MSB TO LSB), 1 = ACTIVE; VIDEO DATA ALWAYS APPEARS IN THE ORDER INDICATED HERE. CONSTANT FOR JOB.
89	1	PROCESSING FLAG 00000000 = RAW DATA 00000001 = PROCESSED DATA FROM COMPUTING SYSTEM
90	22	NUMBER OF CHANNELS IN THIS JOB (EXAMPLE)
91	8	NUMBER OF BITS IN A PICTURE ELEMENT
92-93	1	ADDRESS OF START OF VIDEO DATA GIVES LOCATION OF START OF VIDEO WITHIN SCAN
94-95	0	ADDRESS OF START OF CALIBRATION DATA GIVES LOCATION OF START OF CALIBRATION WITHIN SCAN
96-97	1033*	NUMBER OF VIDEO ELEMENTS PER SCAN WITHIN A SINGLE CHANNEL

*ries due to line straightening

S-192 CYBER 73 LINE STRAIGHTENED DATA TAPE HEADER RECORD (CONTINUED)

BYTES	CONTENTS	DESCRIPTION
98-99	0	NUMBER OF CALIBRATION ELEMENTS PER SCAN WITHIN A SINGLE CHANNEL
100-101	3060	PHYSICAL RECORD SIZE IN BYTES
102	2	NUMBER OF VIDEO CHANNELS IN SECOND AND SUBSEQUENT PHYSICAL RECORDS PER DATA SET
103	0	NUMBER OF PHYSICAL RECORDS PER SCAN PER CHANNEL. THIS FIELD IS USED ONLY WHEN THE NUMBER OF ELEMENTS PER CHANNEL IS GREATER THAN 3K. OTHERWISE IT IS EQUAL TO 0.
104	11	NUMBER OF RECORDS TO MAKE A COMPLETE DATA SET.
105-106	260	LENGTH OF ANCILLARY BLOCK IN BYTES.
107	0	DATA ORDER INDICATOR (EXAMPLE).
108-109	1	START PIXEL NUMBER (EXAMPLE).
110-111	1033	STOP PIXEL NUMBER (EXAMPLE).
112-239*		A_o COEFFICIENT (2 BYTES/CHANNEL).
240-367*		E_o EXPONENT (2 BYTES/CHANNEL).
368-495*		A_1 COEFFICIENT (2 BYTES/CHENNEL).
496-623*		E_1 EXPONENT (2 BYTES/CHANNEL).
624-687		COLOR CODE INFORMATION (TABLE 6.0.1-1).
688-751		SCALE FACTOR (TABLE 6.0.1-1).
752		OFFSET CONSTANT
753	60	WORD SIZE OF GENERATING COMPUTER
754-761	00000520	SHORT WAVELENGTH** END OF SDO 1
762-769	00000560	LONG WAVELENGTH** END OF SDO 1
770-777	00000520	SHORT WAVELENGTH** END OF SDO 2
778-785	00000560	LONG WAVELENGTH** END OF SDO 2
786-793	00000560	SHORT WAVELENGTH** END OF SDO 3
794-801	00000610	LONG WAVELENGTH** END OF SDO 3

*MSB IS A SIGN BIT: 0 = +, 1 = -.
**MILLI-MICRONS - EBCDIC.

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

S-192 CYBER 73 LINE STRAIGHTENED DATA TAPE HEADER RECORD (CONTINUED)

BYTES	CONTENTS	DESCRIPTION
802-809	00000560	SHORT WAVELENGTH* END OF SDO 4
810-817	00000610	LONG WAVELENGTH* END OF SDO 4
818-825	00000620	SHORT WAVELENGTH* END OF SDO 5
826-833	00000670	LONG WAVELENGTH* END OF SDO 5
834-841	00000620	SHORT WAVELENGTH* END OF SDO 6
842-849	00000670	LONG WAVELENGTH* END OF SDO 6
850-857	00000680	SHORT WAVELENGTH* END OF SDO 7
858-865	00000760	LONG WAVELENGTH* END OF SDO 7
866-873	00000680	SHORT WAVELENGTH* END OF SDO 8
874-881	00000760	LONG WAVELENGTH* END OF SDO 8
882-889	00000780	SHORT WAVELENGTH* END OF SDO 9
890-897	00000880	LONG WAVELENGTH* END OF SDO 9
898-905	00000780	SHORT WAVELENGTH* END OF SDO 10
906-913	00000880	LONG WAVELENGTH* END OF SDO 10
914-921	00001550	SHORT WAVELENGTH* END OF SDO 11
922-929	00001750	LONG WAVELENGTH* END OF SDO 11
930-937	00001550	SHORT WAVELENGTH* END OF SDO 12
938-945	00001750	LONG WAVELENGTH* END OF SDO 12
946-953	00002100	SHORT WAVELENGTH* END OF SDO 13
954-961	00002350	LONG WAVELENGTH* END OF SDO 13
962-969	00002100	SHORT WAVELENGTH* END OF SDO 14
970-977	00002350	LONG WAVELENGTH* END OF SDO 14
978-985	00010200	SHORT WAVELENGTH* END OF SDO 15
986-993	00012500	LONG WAVELENGTH* END OF SDO 15
994-1001	00010200	SHORT WAVELENGTH* END OF SDO 16
1002-1009	00012500	LONG WAVELENGTH* END OF SDO 16
1010-1017	00001200	SHORT WAVELENGTH* END OF SDO 17
1018-1025	00001300	LONG WAVELENGTH* END OF SDO 17
1026-1033	00000460	SHORT WAVELENGTH* END OF SDO 18

*MILLI-MICRONS - EBCDIC

S-192 CYBER 73 LINE STRAIGHTENED DATA TAPE HEADER RECORD (CONTINUED)

BYTES	CONTENTS	DESCRIPTION
1034-1041	00000510	LONG WAVELENGTH* END OF SDO 18
1042-1049	00000980	SHORT WAVELENGTH* END OF SDO 19
1050-1057	00001030	LONG WAVELENGTH* END OF SDO 19
1058-1065	00001090	SHORT WAVELENGTH* END OF SDO 20
1066-1073	00001190	LONG WAVELENGTH* END OF SDO 20
1074-1081	00010200	SHORT WAVELENGTH* END OF SDO 21
1082-1089	00012500	LONG WAVELENGTH* END OF SDO 21
1090-1097	00000410	SHORT WAVELENGTH* END OF SDO 22
1098-1105	00000460	LONG WAVELENGTH* END OF SDO 22
1106-1777	0	FILL ZERO
1778	0	NUMBER OF DATA SETS PER PHYSICAL RECORD
1779-1780	0	ADDRESS OF START OF SECOND CALIBRATION WITHIN SCAN. IF THE SCAN HAS ONLY ONE CALIBRATION AREA, THIS FIELD WILL CONTAIN ZEROS.
1781-1782	0	NUMBER OF CALIBRATION ELEMENTS IN THE SECOND CALIBRATION AREA WITHIN THE SCAN IN A SINGLE CHANNEL. IF THE SCAN HAS ONLY ONE CALIBRATION AREA, THIS FIELD WILL CONTAIN ZEROS.
1783	0	CALIBRATION SOURCE INDICATOR LSB = SECOND CALIBRATION AREA LSB + 1 = FIRST CALIBRATION AREA 0 = LOW CALIBRATION SOURCE DATA PRESENT 1 = HIGH CALIBRATION SOURCE DATA PRESENT
1784	0	FILL ZEROS
1785-1786	2	NUMBER OF CHANNELS IN THE FIRST PHYSICAL RECORD OF THE DATA SET.
1787-1788	1033**	TOTAL NUMBER OF ELEMENTS PER SCAN PER CHANNEL.
1789-1790	1	PIXEL SKIP FACTOR (EXAMPLE).
1791-1792	1	SCAN SKIP FACTOR (EXAMPLE).
1793-2184	0	FILL ZEROS

*MILLI-MICRONS - EBCDIC
**SAME VALUE AS IN BYTES 96-97

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

S-192 CYBER 73 LINE STRAIGHTENED DATA TAPE HEADER (CONTINUED)

BYTES	CONTENTS	DESCRIPTION
2185-2550		GENERAL ANNOTATION BYTE ASSIGNMENTS FOR ERIPS (NOT SUPPLIED BY PDP)
2185-2484		USER COMMENTS
2485-2488		FILL ZEROS
2489-2496		LATITUDE OF PIXEL 1 OF REGISTERED IMAGE
2497-2504		LONGITUDE OF LINE 1 OF REGISTERED IMAGE
2505-2512		LATITUDE SCALE FACTOR - DEG/PIXEL
2513-2520		LONGITUDE SCALE FACTOR - DEG/PIXEL
2521-2550		FILL ZEROS
2551-2642		GENERAL ANNOTATION BYTE ASSIGNMENTS FOR THE CYBER 73
2551		BILEVEL WORD NO. 1
		BIT 1 SPARE, SET TO ZERO
		BIT 2 SCAN ANGLE CORRECTION REQUESTED 1 = YES, 0 = NO
		BIT 3 NON-LINEARITY CORRECTION REQUESTED 1 = YES, 0 = NO
		BIT 4 EARTH ROTATION CORRECTION REQUESTED 1 = YES, 0 = NO
		BIT 5 E.U. CONVERSION REQUESTED 1 = YES, 0 = NO
		BIT 6 SPARE
		BIT 7 SPARE
		BIT 8 OMIT SCREENING OUTPUT 1 = YES, 0 = NO
2552		BILEVEL WORD NO. 2
		BIT 1 1 = BAND FORMAT STRAIGHTENED 0 = SDO FORMAT STRAIGHTENED
		BIT 2 LINE STRAIGHTENED DATA REQUESTED (9-TRK) 1 = YES, 0 = NO
		BIT 3 CONICAL DATA REQUESTED (9-TRK) 1 = YES, 0 = NO

S-192 CYBER 73 LINE STRAIGHTENED DATA TAPE HEADER RECORD (CONTINUED)

BYTES	CONTENTS	DESCRIPTION
2551-2642		GENERAL ANNOTATION BYTE ASSIGNMENTS FOR THE CYBER 73 (CONTINUED)
2552		BILEVEL WORD NO. 2 (CONTINUED) BIT 4 LINE STRAIGHTENED DATA REQUESTED (7-TRK) 1 = YES, 0 = NO BIT 5 CONICAL DATA REQUESTED (7-TRK) 1 = YES, 0 = NO BIT 6 CREATE COLOR COMPOSITE IMAGERY 1 = YES, 0 = NO BIT 7 1 = BAND FORMAT CONICAL 0 = SDO FORMAT CONICAL BIT 8 CREATE SINGLE COLOR IMAGERY 1 = YES, 0 = NO
2553-2555		CHANNELS REQUESTED ON STRAIGHTENED OUT EACH CHANNEL WILL BE REPRESENTED BY ONE BIT, WITH THE BITS ARRANGED IN ORDER FROM 1 TO 24 WITH CHANNEL ONE IN THE MSB OF THE FIRST BYTE. A ONE MEANS THE CHANNEL IS REQUIRED, A ZERO MEANS IT IS NOT REQUESTED.
2556-2564		FILL ZEROS
2565-2568		RADIANCE UNIT TAB START TIME (ELEMENT VALUE)* THE GMT REPRESENTING THE TIME AT WHICH THE TAB RUN IS TO BEGIN. THE TIME IS RECORDED IN TENTHS OF MILLISECONDS.
2569-2570		TOTAL NUMBER OF SCAN LINES TO BE PRINTED
2571-2572		INCREMENT FOR SCAN LINE TAB
2573-2575		FILL ZEROS
2576		NUMBER OF COPIES OF CONICAL TAPE REQUIRED
2577		NUMBER OF ELEMENT VALUE TAB COPIES REQUESTED THE NUMBER OF TAB COPIES REQUESTED REPRESENTED IN A BINARY.

*ZEROS IN THE TAB REQUEST START AND STOP TIMES WILL INDICATE THAT NO TAB LISTINGS ARE REQUIRED.

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

S-192 CYBER 73 LINE STRAIGHTENED DATA TAPE HEADER RECORD (CONTINUED)

BYTE NO.	DESCRIPTION
2551-2642	GENERAL ANNOTATION BYTE ASSIGNMENTS FOR THE CYBER 73 (CONTINUED)
2578	NUMBER OF COPIES OF STRAIGHTENED TAPE REQUESTED THE BINARY REPRESENTATION OF THE NUMBER OF TAPE COPIES REQUESTED.
2579	CALIBRATION PROCESSING INTERVAL THE BINARY REPRESENTATION OF THE SCAN LINE INTERVAL
2580-2587	FILL ZEROS
2588	NUMBER OF COPIES OF STATISTICAL TAB REQUESTED
2589	FILL ZEROS
2590-2593	LIMIT SENSE TAB START TIME
2594-2595	TOTAL NUMBER OF SCAN LINES TO BE PROCESSED
2596-2597	FILL ZEROS
2598	NUMBER OF COPIES OF LIMIT SENSE TAB REQUESTED
2599	FILL ZEROS
2600-2603	START TIME OF ELEMENT VALUE VS ELEMENT NUMBER PLOT
2604-2605	TOTAL NUMBER OF SCAN LINES
2606-2607	INCREMENT FOR SCAN LINE PLOT (NUMBER OF SCAN LINES)
2608-2642	FILL ZEROS (END OF CYBER 73 ANNOTATION)
2643-2758	FILL ZEROS

S-192 CYBER 73 LINE STRAIGHTENED DATA TAPE HEADER RECORD (CONTINUED)

BYTE NO.	DESCRIPTION
2759-2940	GENERAL ANNOTATION BYTE ASSIGNMENTS FOR THE PRODUCTION FILM CONVERTER
2759*	n THOUSAND SCAN LINES PER FRAME - BINARY
2760-2789*	JOB ID - EBCDIC
2790-2792*	ALTITUDE IN METERS - BINARY
2793-2794*	GROUND SPEED IN METERS PER SECOND - BINARY
2795*	SCAN TYPE - BINARY 0 = LINEAR 1 = SMOOTHED
2796*	ANGLE OF ARC IN DEGREES - BINARY
2797*	CAMERA - BINARY 0 = 70 mm 1 = 5-INCH
2798*	INPUT DEVICE - BINARY 0 = 9-TRK 1 = HDT
2799*	TRUNCATION - BINARY 0 = LO 1 = HI 2 = NO
2800-2807*	CHANNELS REQUESTED 1 BIT/CHANNEL UP TO 64 TOTAL CHANNELS - BINARY
2808*	PROCESSING MODE - BINARY 0 = SERIALLY 1 = CONCURRENTLY
2809-2873	OVERLAY IMAGE FACTORS - BINARY $A_0, A_1, A_2 \ldots A_n$ WHERE n = CHANNEL NUMBER AND n = 0, IS THE OVERALL FACTOR; THERE IS AN IMPLIED DECIMAL POINT TO LEFT OF THE LEAST SIGNIFICANT DECIMAL DIGIT; IF THE MSB = 1, THE NUMBER IS NEGATIVE; IF = 0, POSITIVE.
2874*	COLOR SELECT - BINARY 0 = NO COLOR 1 = ASSIGNED COLOR 2 = FALSE COLOR

*REQUIRED FOR IMAGE PROCESSING CALCULATIONS.

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

S-192 CYBER 73 LINE STRAIGHTENED DATA TAPE HEADER RECORD (CONTINUED)

BYTE NO.	DESCRIPTION
2759-2940	GENERAL ANNOTATION BYTE ASSIGNMENTS FOR THE PRODUCTION FILM CONVERTER (CONTINUED)
2875*	IMAGE FORMAT - BINARY 0 = SINGLE IMAGE 1 = OVERLAY IMAGES 2 = ABUT IMAGES 3 = OFFSET IMAGES
2876*	REPEAT OF PIXELS PER SCAN - BINARY 0 = NONE 1 = 1 REPEAT 2 = 2 REPEAT n = n REPEAT
2877	REPEAT OF SCAN - BINARY 0 = NONE 1 = 1 REPEAT 2 = 2 REPEAT n = N REPEAT
2878-2881	PARTIAL SCAN - BINARY ZEROS TO INDICATE FULL SCAN BYTE 2878-2879 = FROM PIXEL NO. BYTE 2880-2881 = TO PIXEL NO.
2882-2883	SENSOR SCAN RATE IN SCANS/SECOND - BINARY
2884	PIXEL SIZE - BINARY
2885-2886	ANGLE OF DRIFT - BINARY BYTE 2885 = ± INTEGER BYTE 2886 = FRACTION
2887-2940	FILL ZEROS
2941-3000	ITITLE - USER DESIGNATED IDENTIFICATION
3001-3060	FILL ZEROS, MAKES THE RECORD AN INTEGRAL NUMBER OF COMPUTER WORDS. THESE BYTES MUST NEVER CONTAIN DATA.

*REQUIRED FOR IMAGE PROCESSING CALCULATIONS.

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

S192 CYBER 73 LINE STRAIGHTENED DATA TAPE ANCILLARY BLOCK

BLOCK BYTE NO.*	CONTENT
1-4	TIME AT THE START OF THIS DATA SET, RECORDED IN TENTHS OF MILLISECONDS
5-68	CHANNEL STATUS CHANNELS 1 THRU 22 = 1 OR 0 DEPENDING ON STATUS; CHANNELS 23 THRU 64 = 0 (PAGE 6.0.1-11)
69-70	SCAN LINE NUMBER
71-74	GEODETIC LATITUDE - SPACECRAFT NADIR BYTE 71 = SIGN + = ALL ZEROS, - = 00000001 BYTE 72 = DEGREES BYTE 73 = MINUTES BYTE 74 = QUARTER SECONDS
75-78	LONGITUDE - SPACECRAFT NADIR BYTE 75 = SIGN + = ALL ZEROS, - = 00000001 BYTE 76 = DEGREES BYTE 77 = MINUTES BYTE 78 = QUARTER SECONDS
79-82	ROLL ANGLE (+ = RIGHT WING UP) BYTE 79 = SIGN + = ALL ZEROS, - = 00000001 BYTE 80 = DEGREES BYTE 81 = MINUTES BYTE 82 = QUARTER SECONDS
83-86	PITCH ANGLE (+ = NOSE UP) BYTE 83 = SING + = ALL ZEROS, - = 00000001 BYTE 84 = DEGREES BYTE 85 = MINUTES BYTE 86 = QUARTER SECONDS
87-90	YAW ANGLE (+ = NOSE TOWARD RIGHT WING) BYTE 87 = SIGN + = ALL ZEROS, - = 00000001 BYTE 88 = DEGREES BYTE 89 = MINUTES BYTE 90 = QUARTER SECONDS
91-94	SUN ANGLE BYTE 91 = SIGN + = ALL ZEROS, - = 00000001 BYTE 92 = DEGREES BYTE 93 = MINUTES BYTE 94 = QUARTER SECONDS
95-118	LATITUDE AND LONGITUDE OF IMAGE FOOTPRINT
95-98	LATITUDE OF CENTER OF FIRST PIXEL IN SCAN ARC BYTE 95 = SIGN + = ALL ZEROS, - = 00000001 BYTE 96 = DEGREES BYTE 97 = MINUTES BYTE 98 = QUARTER SECONDS

*DATA RECORD BYTES 1 - 2 CONTAIN PHYSICAL RECORD COUNTER.

S192 CYBER 73 LINE STRAIGHTENED DATA TAPE ANCILLARY BLOCK (CONTINUED)

BLOCK BYTE NO.	CONTENT
95-118 (continued)	
99-102	LONGITUDE OF CENTER OF FIRST PIXEL IN SCAN ARC BYTE 99 = SIGN + = ALL ZEROS, - = 00000001 BYTE 100 = DEGREES BYTE 101 = MINUTES BYTE 102 = QUARTER SECONDS
103-106	LATITUDE OF CENTER OF CENTER PIXEL IN SCAN ARC BYTE 103 = SIGN + = ALL ZEROS, - = 00000001 BYTE 104 = DEGREES BYTE 105 = MINUTES BYTE 106 = QUARTER SECONDS
107-110	LONGITUDE OF CENTER OF CENTER PIXEL IN SCAN ARC BYTE 107 = SIGN + = ALL ZEROS, - = 00000001 BYTE 108 = DEGREES BYTE 109 = MINUTES BYTE 110 = QUARTER SECONDS
111-114	LATITUDE OF CENTER OF LAST PIXEL IN SCAN ARC BYTE 111 = SIGN + = ALL ZEROS, - = 00000001 BYTE 112 = DEGREES BYTE 113 = MINUTES BYTE 114 = QUARTER SECONDS
115-118	LONGITUDE OF CENTER OF LAST PIXEL IN SCAN ARC BYTE 115 = SIGN + = ALL ZEROS, - = 00000001 BYTE 116 = DEGREES BYTE 117 = MINUTES BYTE 118 = QUARTER SECONDS
119-202	S192 HOUSEKEEPING PARAMETERS IN EBCDIC
119-122	A001-192 IN TENTHS OF °K
123-126	A002-192 IN TENTHS OF °K
127-130	A003-192 IN TENTHS OF MILLIAMPS
131-134	A004-192 IN TENTHS OF °K
135-138	A005-192 IN TENTHS OF MILLIAMPS
139-142	A006-192 IN TENTHS OF °K
143-146	A007-192 IN TENTHS OF °K
147-150	A008-192 IN TENTHS OF °K
151-154	A009-192 IN TENTHS OF °K
155-158	A010-192 IN TENTHS OF °K
159-162	A011-192 IN TENTHS OF °K
163-166	A012-192 IN TENTHS OF °K
167-170	A013-192 IN TENTHS OF °K
171-174	A014-192 IN TENTHS OF °K
175-178	A015-192 IN TENTHS OF °K
179-182	A016-192 IN TENTHS OF °K

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

S192 CYBER 73 LINE STRAIGHTENED DATA TAPE ANCILLARY BLOCK (CONTINUED)

BLOCK BYTE NO.	CONTENT
119-202 (continued)	
183-186	A017-192 IN TENTHS OF ^{o}K
187-190	A018-192 IN TENTHS OF ^{o}K
191-194	A019-192 IN TENTHS OF ^{o}K
195-198	A020-192 IN TENTHS OF ^{o}K
199-202	A021-192 IN TENTHS OF MICROSECONDS
203	SCAN SHIFT DELAY FLAG 0 (00000000) = NO SHIFT +1 (00000001) = MOVE TO RIGHT -1 (11111111) = MOVE TO LEFT
204	TO BE DEFINED
205-206	AXIMUTH (HEADING) IN DEGREES
207-248	FILL ZERO
249-252	GROUND VELOCITY IN METERS PER SECOND
253-256	ALTITUDE IN METERS
257-260	ORBITAL INCLINATION ANGLE IN QUARTER ARC SECONDS

S-192 CYBER 73 LINE STRAIGHTENED DATA TAPE RECORDS*

RECORD 1 OF DATA SET

BYTE NO.	DESCRIPTION
1-2	FRAME COUNTER
3-262	ANCILLARY BLOCK
263-2328	VIDEO BLOCK
263-1295	1033 SAMPLES OF VIDEO FROM SDO 1 (EXAMPLE)
1296-2328	1033 SAMPLES OF VIDEO FROM SDO 2 (EXAMPLE)
2329-3060	FILL ZEROS (EXAMPLE)

RECORD 2 OF DATA SET

BYTE NO.	DESCRIPTION
1-2	FRAME COUNTER
3-2068	VIDEO BLOCK (CONTINUED)
3-1035	1033 SAMPLES OF VIDEO FROM SDO 3 (EXAMPLE)
1036-2068	1033 SAMPLES OF VIDEO FROM SDO 4 (EXAMPLE)
2069-3060	FILL ZEROS (EXAMPLE)

RECORD 3 OF DATA SET

BYTE NO.	DESCRIPTION
1-2	FRAME COUNTER
3-2068	VIDEO BLOCK (CONTINUED)
3-1035	1033 SAMPLES OF VIDEO FROM SDO 5 (EXAMPLE)
1036-2068	1033 SAMPLES OF VIDEO FROM SDO 6 (EXAMPLE)
2069-3060	FILL ZEROS (EXAMPLE)

RECORD 4 OF DATA SET

BYTE NO.	DESCRIPTION
1-2	FRAME COUNTER
3-2068	VIDEO BLOCK (CONTINUED)
3-1035	1033 SAMPLES OF VIDEO FROM SDO 7 (EXAMPLE)
1036-2068	1033 SAMPLES OF VIDEO FROM SDO 8 (EXAMPLE)
2069-3060	FILL ZEROS (EXAMPLE)

RECORD 5 OF DATA SET

BYTE NO.	DESCRIPTION
1-2	FRAME COUNTER
3-2068	VIDEO BLOCK (CONTINUED)
3-1035	1033 SAMPLES OF VIDEO FROM SDO 9 (EXAMPLE)
1036-2068	1033 SAMPLES OF VIDEO FROM SDO 10 (EXAMPLE)
2069-3060	FILL ZEROS (EXAMPLE)

*VIDEO SAMPLES FROM SDO'S 15, 16 AND 21 ARE IN WATTS/CM^2/STER/MICRON; FROM ALL OTHER SDO'S IN MILLIWATTS/CM^2/STER/MICRON.

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

S-192 CYBER 73 LINE STRAIGHTENED DATA TAPE RECORDS (CONTINUED)

RECORD 6 OF DATA SET

BYTE NO.	DESCRIPTION
1-2	FRAME COUNTER
3-2068	VIDEO BLOCK (CONTINUED)
3-1035	1033 SAMPLES OF VIDEO FROM SDO 11 (EXAMPLE)
1036-2068	1033 SAMPLES OF VIDEO FROM SDO 12 (EXAMPLE)
2069-3060	FILL ZEROS (EXAMPLE)

RECORD 7 OF DATA SET

BYTE NO.	DESCRIPTION
1-2	FRAME COUNTER
3-2068	VIDEO BLOCK (CONTINUED)
3-1035	1033 SAMPLES OF VIDEO FROM SDO 13 (EXAMPLE)
1036-2068	1033 SAMPLES OF VODEO FROM SDO 14 (EXAMPLE)
2069-3060	FILL ZEROS (EXAMPLE)

RECORD 8 OF DATA SET

BYTE NO.	DESCRIPTION
1-2	FRAME COUNTER
3-2068	VIDEO BLOCK (CONTINUED)
3-1035	1033 SAMPLES OF VIDEO FROM SDO 15 (EXAMPLE)
1036-2068	1033 SAMPLES OF VIDEO FROM SDO 16 (EXAMPLE)
2069-3060	FILL ZEROS (EXAMPLE)

RECORD 9 OF DATA SET

BYTE NO.	DESCRIPTION
1-2	FRAME COUNTER
3-2068	VIDEO BLOCK (CONTINUED)
3-1035	1033 SAMPLES OF VIDEO FROM SDO 17 (EXAMPLE)
1036-2068	1033 SAMPLES OF VIDEO FROM SDO 18 (EXAMPLE)
2069-3060	FILL ZEROS (EXAMPLE)

S-192 CYBER 73 LINE STRAIGHTENED DATA TAPE RECORDS (CONTINUED)

RECORD 10 OF DATA SET

BYTE NO.	DESCRIPTION
1-2	FRAME COUNTER
3-2068	VIDEO BLOCK (CONTINUED)
3-1035	1033 SAMPLES OF VIDEO FROM SDO 19 (EXAMPLE)
1036-2068	1033 SAMPLES OF VIDEO FROM SDO 20 (EXAMPLE)
2069-3060	FILL ZEROS (EXAMPLE)

RECORD 11 OF DATA SET

BYTE NO.	DESCRIPTION
1-2	FRAME COUNTER
3-2068	VIDEO BLOCK (CONTINUED)
3-1035	1033 SAMPLES OF VIDEO FROM SDO 21 (EXAMPLE)
1036-2068	1033 SAMPLES OF VIDEO FROM SDO 22 (EXAMPLE)
2069-3060	FILL ZEROS (EXAMPLE)

REPRODUCIBILITY OF THI
ORIGINAL PAGE IS POOR

6.1.3 S192 CYBER 73 Processed Data Tapes

This section describes the Processed Data Tapes which are output by the CYBER 73 at JSC. They consist of Conical Scan Line Data Tapes and Scan Line Straightened Data Tapes.

The Conical Scan Line Data Tape contains data which has undergone scan angle and non-linearity corrections, calibration, and scaling.

The Scan Line Straightened Data Tape format is very similar to the Conical Scan Line tape, but the data blocks do not contain either calibration or subframe status.

S192 CONICAL SCAN LINE DATA TAPE HEADER RECORD

(The format of this header record is identical to that of the PDP-CYBER Interface Tape Header Record.) Contents of certain bytes are different however. They are described as follows:

RECORD BYTE NO.	PDP-CYBER INTERFACE CONTENT	CONICAL SCAN LINE CONTENT
100-101	2880	3060
1787-1788	1260	1258

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

S192 CONICAL SCAN LINE DATA TAPE ANCILLARY BLOCK

(The format and content of this ancillary block is identical to that of the PDP-CYBER Interface Tape ancillary block.)

S192 CONICAL SCAN LINE DATA TAPE*

RECORD 1 OF DATA SET

BYTE NO.	DESCRIPTION
1-2	FRAME COUNTER
3-2778	VIDEO BLOCK
3-262	ANCILLARY BLOCK
263-280	18 SAMPLES OF CALIBRATION FROM SDO 1
281-1520	1240 SAMPLES OF VIDEO FROM SDO 1
1521-1538	18 SAMPLES OF CALIBRATION FROM SDO 2
1539-2778	1240 SAMPLES OF VIDEO FROM SDO 2
2779-3060	FILL ZEROS

RECORD 2 OF DATA SET

BYTE NO.	DESCRIPTION
1-2	FRAME COUNTER
3-2518	VIDEO BLOCK (CONTINUED)
3-20	18 SAMPLES OF CALIBRATION FROM SDO 3
21-1260	1240 SAMPLES OF VIDEO FROM SDO 3
1261-1278	18 SAMPLES OF CALIBRATION FROM SDO 4
1279-2518	1240 SAMPLES OF VIDEO FROM SDO 4
2519-3060	FILL ZEROS

RECORD 3 OF DATA SET

BYTE NO.	DESCRIPTION
1-2	FRAME COUNTER
3-2518	VIDEO BLOCK (CONTINUED)
3-20	18 SAMPLES OF CALIBRATION FROM SDO 5
21-1260	1240 SAMPLES OF VIDEO FROM SDO 5
1261-1278	18 SAMPLES OF CALIBRATION FROM SDO 6
1279-2518	1240 SAMPLES OF VIDEO FROM SDO 6
2519-3060	FILL ZEROS

*VIDEO SAMPLES FROM SDO'S 15, 16 AND 21 ARE IN WATTS/CM^2/STER/MICRON; FROM ALL OTHER SDO'S IN MILLIWATTS/CM^2/STER/MICRON.

REPRODUCIBILITY OF THE ORIGINAL PAGE IS POOR

S192 CONICAL SCAN LINE DATA TAPE (CONTINUED)

RECORD 4 OF DATA SET

BYTE NO.	DESCRIPTION
1-2	FRAME COUNTER
3-2518	VIDEO BLOCK (CONTINUED)
3-20	18 SAMPLES OF CALIBRATION FROM SDO 7
21-1260	1240 SAMPLES OF VIDEO FROM SDO 7
1261-1278	18 SAMPLES OF CALIBRATION FROM SDO 8
1279-2518	1240 SAMPLES OF VIDEO FROM SDO 8
2519-3060	FILL ZEROS

RECORD 5 OF DATA SET

BYTE NO.	DESCRIPTION
1-2	FRAME COUNTER
3-2518	VIDEO BLOCK (CONTINUED)
3-20	18 SAMPLES OF CALIBRATION FROM SDO 9
21-1260	1240 SAMPLES OF VIDEO FROM SDO 9
1261-1278	18 SAMPLES OF CALIBRATION FROM SDO 10
1279-2518	1240 SAMPLES OF VIDEO FROM SDO 10
2519-3060	FILL ZEROS

RECORD 6 OF DATA SET

BYTE NO.	DESCRIPTION
1-2	FRAME COUNTER
3-2518	VIDEO BLOCK (CONTINUED)
3-20	18 SAMPLES OF CALIBRATION FROM SDO 11
21-1260	1240 SAMPLES OF VIDEO FROM SDO 11
1261-1278	18 SAMPLES OF CALIBRATION FROM SDO 12
1279-2518	1240 SAMPLES OF VIDEO FROM SDO 12
2519-3060	FILL ZEROS

RECORD 7 OF DATA SET

BYTE NO.	DESCRIPTION
1-2	FRAME COUNTER
3-2518	VIDEO BLOCK (CONTINUED)
3-20	18 SAMPLES OF CALIBRATION FROM SDO 13
21-1260	1240 SAMPLES OF VIDEO FROM SDO 13
1261-1278	18 SAMPLES OF CALIBRATION FROM SDO 14
1279-2518	1240 SAMPLES OF VIDEO FROM SDO 14
2519-3060	FILL ZEROS

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

S192 CONICAL SCAN LINE DATA TAPE (CONTINUED)

RECORD 8 OF DATA SET

BYTE NO.	DESCRIPTION
1-2	FRAME COUNTER
3-2518	VIDEO BLOCK (CONTINUED)
3-20	18 SAMPLES OF CALIBRATION* FROM SDO 15
21-1260	1240 SAMPLES OF VIDEO FROM SDO 15
1261-1278	18 SAMPLES OF CALIBRATION* FROM SDO 16
1279-2518	1240 SAMPLES OF VIDEO FROM SDO 16
2519-3060	FILL ZEROS

RECORD 9 OF DATA SET

BYTE NO.	DESCRIPTION
1-2	FRAME COUNTER
3-2518	VIDEO BLOCK (CONTINUED)
3-20	18 SAMPLES OF CALIBRATION FROM SDO 17
21-1260	1240 SAMPLES OF VIDEO FROM SDO 17
1261-1278	18 SAMPLES OF CALIBRATION FROM SDO 18
1279-2518	1240 SAMPLES OF VIDEO FROM SDO 18
2519-3060	FILL ZEROS

RECORD 10 OF DATA SET

BYTE NO.	DESCRIPTION
1-2	FRAME COUNTER
3-2518	VIDEO BLOCK (CONTINUED)
3-20	18 SAMPLES OF CALIBRATION FROM SDO 19
21-1260	1240 SAMPLES OF VIDEO FROM SDO 19
1261-1278	18 SAMPLES OF CALIBRATION FROM SDO 20
1279-2518	1240 SAMPLES OF VIDEO FROM SDO 20
2519-3060	FILL ZEROS

* SAMPLES 1-6 AND SAMPLES 13-18 AS RECORDED ON THE 28-TRACK EREP TAPE ARE INTERCHANGED FOR THIS OUTPUT TAPE SO THAT HI CAL AND LO CAL WILL BE IN THE SAME ORDER FOR SDO'S 15, 16, AND 21 AS THEY ARE FOR THE OTHER SDO'S.

S192 CONICAL SCAN LINE DATA TAPE (CONTINUED)

RECORD 11 OF DATA SET

BYTE NO.	DESCRIPTION
1-2	FRAME COUNTER
3-2518	VIDEO BLOCK (CONTINUED)
3-20	18 SAMPLES OF CALIBRATION* FROM SDO 21
21-1260	1240 SAMPLES OF VIDEO FROM SDO 21
1261-1278	18 SAMPLES OF CALIBRATION FROM SDO 22
1279-2518	1240 SAMPLES OF VIDEO FROM SDO 22
2519-3060	FILL ZEROS

* SAMPLES 1-6 AND 13-18 AS RECORDED ON THE 28-TRACK EREP TAPE ARE INTERCHANGED FOR THIS OUTPUT TAPE SO THAT HI CAL AND LO CAL WILL BE IN THE SAME ORDER FOR SDO'S 15, 16, AND 21 AS THEY ARE FOR THE OTHER SDO'S.

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

APPENDIX K - RECORDED WAVELENGTHS FOUND ON S-191 DATA TAPES AND CONTENTS OF REORDERED S-191 GSFC VERSION TAPES

SKYLAB Wavelengths (Microns) Which Are Found in S-191 Tapes
Received June 24, 1974

(1)	.400	A21	(28)	.620	B27	(55)	.890	A56
(2)	.405	A22	(29)	.630	B28	(56)	.900	A57
(3)	.410	A23	(30)	.640	B29	(57)	.915	B37
(4)	.415	A24	(31)	.650	B30	(58)	.930	B38
(5)	.420	A25	(32)	.660	B31	(59)	.945	B39
(6)	.425	A26	(33)	.670	B32	(60)	.960	B40
(7)	.430	A27	(34)	.680	B33	(61)	.975	B41
(8)	.435	A28	(35)	.690	B34	(62)	.990	B42
(9)	.440	A29	(36)	.700	B35	(63)	1.10	B43
(10)	.445	A30	(37)	.710	B36	(64)	1.12	B44
(11)	.450	A31	(38)	.720	A39	(65)	1.14	B45
(12)	.460	A32	(39)	.730	A40	(66)	1.16	B46
(13)	.470	A33	(40)	.740	A41	(67)	1.18	B47
(14)	.480	A34	(41)	.750	A42	(68)	1.20	B48
(15)	.490	A35	(42)	.760	A43	(69)	1.22	B49
(16)	.500	A36	(43)	.770	A44	(70)	1.24	B50
(17)	.510	A37	(44)	.780	A45	(71)	1.26	B51
(18)	.520	A38	(45)	.790	A46	(72)	1.28	B52
(19)	.530	B18	(46)	.800	A47	(73)	1.30	B53
(20)	.540	B19	(47)	.810	A48	(74)	1.32	B54
(21)	.550	B20	(48)	.820	A49	(75)	1.34	B55
(22)	.560	B21	(49)	.830	A50	(76)	1.36	B56
(23)	.570	B22	(50)	.840	A51	(77)	1.38	A1
(24)	.580	B23	(51)	.850	A52	(78)	1.40	A2
(25)	.590	B24	(52)	.860	A53	(79)	1.44	A3
(26)	.600	B25	(53)	.870	A54	(80)	1.48	A4
(27)	.610	B26	(54)	.880	A55	(81)	1.50	A5

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

SKYLAB Wavelengths (Microns) Which Are Found in S-191 Tapes
Received June 24, 1974 (Continued)

(82)	1.52	A6	(109)	2.32	B13	(136)	8.2	B98
(83)	1.54	A7	(110)	2.36	B14	(137)	8.3	B99
(84)	1.56	A8	(111)	2.40	B15	(138)	8.4	B100
(85)	1.58	A9	(112)	2.44	B16	(139)	8.5	B101
(86)	1.60	A10	(113)	2.48	B17	(140)	8.6	B102
(87)	1.62	A11	(114)	6.0	A93	(141)	8.7	B103
(88)	1.64	A12	(115)	6.1	A94	(142)	8.8	B104
(89)	1.66	A13	(116)	6.2	A95	(143)	8.9	B105
(90)	1.68	A14	(117)	6.3	A96	(144)	9.0	B106
(91)	1.70	A15	(118)	6.4	A97	(145)	9.1	B107
(92)	1.72	A16	(119)	6.5	A98	(146)	9.2	B108
(93)	1.74	A17	(120)	6.6	A99	(147)	9.3	A58
(94)	1.76	A18	(121)	6.7	A100	(148)	9.4	A59
(95)	1.78	A19	(122)	6.8	A101	(149)	9.5	A60
(96)	1.80	A20	(123)	6.9	A102	(150)	9.6	A61
(97)	1.84	B1	(124)	7.0	A103	(151)	9.7	A62
(98)	1.88	B2	(125)	7.1	A104	(152)	9.8	A63
(99)	1.92	B3	(126)	7.2	A105	(153)	9.9	A64
(100)	1.96	B4	(127)	7.3	A106	(154)	10.	A65
(101)	2.0	B5	(128)	7.4	A107	(155)	10.1	A66
(102)	2.04	B6	(129)	7.5	A108	(156)	10.2	A67
(103)	2.08	B7	(130)	7.6	A109	(157)	10.3	A68
(104)	2.12	B8	(131)	7.7	B93	(158)	10.4	A69
(105)	2.16	B9	(132)	7.8	B94	(159)	10.5	A70
(106)	2.20	B10	(133)	7.9	B95	(160)	10.6	A71
(107)	2.24	B11	(134)	8.0	B96	(161)	10.7	A72
(108)	2.28	B12	(135)	8.1	B97	(162)	10.8	A73

SKYLAB Wavelengths (Microns) Which Are Found in S-191 Tapes
Received June 24, 1974 (Continued)

(163)	10.9	A74
(164)	11.0	B57
(165)	11.1	B58
(166)	11.2	B59
(167)	11.3	B60
(168)	11.4	B61
(169)	11.5	B62
(170)	11.6	B63
(171)	11.7	B64
(172)	11.8	B65
(173)	11.9	B66
(174)	12.0	B67
(175)	12.1	B68
(176)	12.2	B69
(177)	12.3	B70
(178)	12.4	B71
(179)	12.5	B72
(180)	12.6	B73
(181)	12.7	B74
(182)	12.8	A75
(183)	12.9	A76
(184)	13.0	A77
(185)	13.1	A78
(186)	13.2	A79
(187)	13.3	A80
(188)	13.4	A81
(189)	13.5	A82
(190)	13.6	A83
(191)	13.7	A84
(192)	13.8	A85
(193)	13.9	A86
(194)	13.95	A87
(195)	14.00	A88
(196)	14.05	A89
(197)	14.10	A90
(198)	14.15	A91
(199)	14.20	A92
(200)	14.25	B75
(201)	14.30	B76
(202)	14.35	B77
(203)	14.40	B78
(204)	14.45	B79
(205)	14.50	B80
(206)	14.55	B81
(207)	14.60	B82
(208)	14.65	B83
(209)	14.70	B84
(210)	14.75	B85
(211)	14.80	B86
(212)	14.90	B87
(213)	15.0	B88
(214)	15.1	B89
(215)	15.2	B90
(216)	15.3	B91
(217)	15.4	B92

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

SKYLAB Wavelengths (Microns) Which Are Found in S-191 Tapes
Received January 1975

(1)	.400	A21	(28)	.620	B30	(55)	.920	A58
(2)	.405	A22	(29)	.630	B31	(56)	.940	A59
(3)	.410	A23	(30)	.640	B32	(57)	.960	A60
(4)	.415	A24	(31)	.650	B33	(58)	.980	B41
(5)	.420	A25	(32)	.660	B34	(59)	1.00	B42
(6)	.425	A26	(33)	.670	B35	(60)	1.02	B43
(7)	.430	A27	(34)	.680	B36	(61)	1.04	B44
(8)	.435	A28	(35)	.690	B37	(62)	1.06	B45
(9)	.440	A29	(36)	.700	B38	(63)	1.08	B46
(10)	.445	A30	(37)	.710	B39	(64)	1,10	B47
(11)	.450	A31	(38)	.720	A41	(65)	1.12	B48
(12)	.460	A32	(39)	.730	A42	(66)	1.14	B49
(13)	.470	A33	(40)	.740	A43	(67)	1.16	B50
(14)	.480	A34	(41)	.750	A44	(68)	1.18	B51
(15)	.490	A35	(42)	.760	A45	(69)	1.20	B52
(16)	.500	A36	(43)	.770	A46	(70)	1.22	B53
(17)	.510	A37	(44)	.780	A47	(71)	1.24	B54
(18)	.520	A38	(45)	.790	A48	(72)	1.26	B55
(19)	.530	B21	(46)	.800	A49	(73)	1.28	B56
(20)	.540	B22	(47)	.810	A50	(74)	1.30	B57
(21)	.550	B23	(48)	.820	A51	(75)	1.32	B58
(22)	.560	B24	(49)	.830	A52	(76)	1.34	B59
(23)	.570	B25	(50)	.840	A53	(77)	1.36	B60
(24)	.580	B26	(51)	.850	A54	(78)	1.38	A1
(25)	.590	B27	(52)	.860	A55	(79)	1.40	A2
(26)	.600	B28	(53)	.880	A56	(80)	1.44	A3
(27)	.610	B29	(54)	.900	A57	(81)	1.48	A4

SKYLAB Wavelengths (Microns) Which Are Found in S-191 Tapes Received January 1975 (Continued)

(82)	1.50	A5	(109)	2.28	B12	(136)	8.10	B105
(83)	1.52	A6	(110)	2.32	B13	(137)	8.20	B106
(84)	1.54	A7	(111)	2.36	B14	(138)	8.30	B107
(85)	1.56	A8	(112)	2.40	B15	(139)	8.40	B108
(86)	1.58	A9	(113)	2.44	B16	(140)	8.50	B109
(87)	1.60	A10	(114)	2.48	B17	(141)	8.60	B110
(88)	1.62	A11	(115)	6.00	A101	(142)	8.70	B111
(89)	1.64	A12	(116)	6.10	A102	(143)	8.80	B112
(90)	1.66	A13	(117)	6.20	A103	(144)	8.90	B113
(91)	1.68	A14	(118)	6.30	A104	(145)	9.00	B114
(92)	1.70	A15	(119)	6.40	A105	(146)	9.10	B115
(93)	1.72	A16	(120)	6.50	A106	(147)	9.20	B116
(94)	1.74	A17	(121)	6.60	A107	(148)	9.30	A61
(95)	1.76	A18	(122)	6.70	A108	(149)	9.40	A62
(96)	1.78	A19	(123)	6.80	A109	(150)	9.50	A63
(97)	1.80	A20	(124)	6.90	A110	(151)	9.60	A64
(98)	1.84	B1	(125)	7.00	A111	(152)	9.70	A65
(99)	1.88	B2	(126)	7.10	A112	(153)	9.80	A66
(100)	1.92	B3	(127)	7.20	A113	(154)	9.90	A67
(101)	1.96	B4	(128)	7.30	A114	(155)	10.00	A68
(102)	2.00	B5	(129)	7.40	A115	(156)	10.10	A69
(103)	2.04	B6	(130)	7.50	A116	(157)	10.20	A70
(104)	2.08	B7	(131)	7.60	A117	(158)	10.30	A71
(105)	2.12	B8	(132)	7.70	B101	(159)	10.40	A72
(106)	2.16	B9	(133)	7.80	B102	(160)	10.50	A73
(107)	2.20	B10	(134)	7.90	B103	(161)	10.60	A74
(108)	2.24	B11	(135)	8.00	B104	(162)	10.70	A75

SKYLAB Wavelengths (Microns) Which Are Found in S-191 Tapes
Received January 1975 (Continued)

(163)	10.80	A76	(191)	13.60	A89
(164)	10.90	A77	(192)	13.70	A90
(165)	11.00	B61	(193)	13.80	A91
(166)	11.10	B62	(194)	13.90	A92
(167)	11.20	B63	(195)	13.95	A93
(168)	11.30	B64	(196)	14.00	A94
(169)	11.40	B65	(197)	14.05	A95
(170)	11.50	B66	(198)	14.10	A96
(171)	11.60	B67	(199)	14.15	A97
(172)	11.70	B68	(200)	14.20	A18
(173)	11.80	B69	(201)	14.25	B81
(174)	11.90	B70	(202)	14.30	B82
(175)	12.00	B71	(203)	14.35	B83
(176)	12.10	B72	(204)	14.40	B84
(177)	12.20	B73	(205)	14.45	B85
(178)	12.30	B74	(206)	14.50	B86
(179)	12.40	B75	(207)	14.55	B87
(180)	12.50	B76	(208)	14.60	B88
(181)	12.60	B77	(209)	14.65	B89
(182)	12.70	B78	(210)	14.70	B90
(183)	12.80	A81	(211)	14.75	B91
(184)	12.90	A82	(212)	14.80	B92
(185)	13.00	A83	(213)	14.90	B93
(186)	13.10	A84	(214)	15.00	B94
(187)	13.20	A85	(215)	15.10	B95
(188)	13.30	A86	(216)	15.20	B96
(189)	13.40	A87	(217)	15.30	B97
(190)	13.50	A88	(218)	15.40	B98

GSFC S-191 Data Tapes

The GSFC version of Houston JSC's S-191 data tape is a 9-track, density 3, 1600 bpi tape with record form VBS, a logical record size of 1452 bytes, and block size of 29044 bytes. The logical record size of 1452 bytes includes the 4 bytes (needed in IBM VBS) to contain the size of that record. Also, the 29044 bytes include the 4 bytes needed to contain the actual block size (IBM "block size" is sometimes called "physical record size").

To be explicit, if X is the number of bytes needed to contain the record, Y = X + 4 is the logical record size, and Z = N * Y + 4 is the block size, where N is the number of logical records in the file, the value of X is 1448 bytes, and the value of N is 20.

In each case, the 4 bytes indicating record lengths precede that record, so that the first 32 bits (where the record 16 bits are zeros) of the physical block contain the block size. This is followed by the logical records of 32 bits (where the second 16 bits contain IBM indicators as to whether the record is continued, "spill over" from a previous block, etc.).

Since the IBM word size is 32 bits, or 4 bytes, the size of the output array is clearly 362 words. The contents of that array are listed below:

Halfword Integer I*2 (16 bits, or 2 bytes, each)	Floating Point Real R*4 (32 bits, or 4 bytes, each)	Bytes	Description	
1	-	1-2	Month	Time data was taken
2	-	3-4	Day	
3	-	5-6	Year	
4	-	7-8	Hour	
5	-	9-10	Minute	
6	-	11-12	Second	

Halfword Integer I*2 (16 bits, or 2 bytes, each)	Floating Point Real R*4 (32 bits, or 4 bytes, each)	Bytes	Description
-	4	13-16	Latitude (deg)
-	5	17-20	Longitude (deg + w)
-	6	21-24	Altitude (km)
-	7	25-28	Yaw (deg)
-	8	29-32	Pitch (deg)
-	9	33-36	Roll (deg)
-	10	37-40	Sun zenith angle (deg)
-	11-14	41-56	Four pointing angle (deg)
-	15-188	57-752	174 wavelengths (in order) (microns)
-	189-362	753-1448	174 radiance values (in corresponding order) (watts/cm^2/ micron/ster)

Output Array of NWT191 Found in GSFC Version S-191 Tapes
Received January 1975

Halfword Integer I*2 (16 bits, or 2 bytes, each)	Floating Point Real R*4 (32 bits, or 4 bytes, each)	Bytes	Description	
1	-	1-2	Month	Time data was taken
2	-	3-4	Day	
3	-	5-6	Year	
4	-	7-8	Hour	
5	-	9-10	Minute	
6	-	11-12	Second	
-	4	13-16	Latitude	(deg)
-	5	17-20	Longtitude	(deg)
-	6	21-24	Altitude	(km)
-	7	25-28	Yaw	(deg)
-	8	29-32	Pitch	(deg)
-	9	33-36	Roll	(deg)
-	10	37-40	Sun zenith angle	(deg)
-	11-14	41-56	Four pointing angles	(deg)
-	15-232	57-928	218 wavelengths (in order)	(microns)
-	233-450	929-1800	218 Radiance values (in corresponding order)	(watts/cm^2 micron/ster)

APPENDIX L - COMMON TRACES

SKYLAB - Originated Common Trace

Main Program Package Name: T91

COMMON Name: VALUES

Variable[1]	Type	Dimension	Initialized or Reset in Subroutine	Referenced in Subroutine
ITIMPT	I*4	3,500	T91 (Main)	PTING1
PTING	R*4	4,500	T91 (Main)	PTING1
ITIMZN	I*4	3,500	T91 (Main)	ZENPTN
ZENVL	R*4	500	T91 (Main)	ZENPTN
NOPT	I*4	-	T91 (Main)	PTING1
NOZE	I*4	-	T91 (Main)	ZENPTN
ISDTPT	L*4	-	T91 (Main)	PTING1
ISDTZN	L*4	-	T91 (Main)	ZENPTN

[1] For descriptions, see Section 2.2.1.4.

SKYLAB - Originated Common Trace

Main Program Package Name: QUICK3

COMMON Name: VALUE

Variable[1]	Type	Dimension	Initialized or Reset in Subroutine	Referenced in Subroutine
LR3	I*4	-	Main (QUICK3)	TEMBRH (ver 2) REFLE1 (ver 2) CONTOU (ver 2) CLUTHK (ver 2)
LC3	I*4	-	Main (QUICK3)	TEMBRH (ver 2) REFLE1 (ver 2) CONTOU (ver 2) CLUTHK (ver 2)
TBB	R*4	(200, 200)	Main (QUICK3) TEMBRH(ver 2) REFLE1(ver 2) CLUTHK(ver 2)	TEMBRH (ver 2) REFLE1 (ver 2) CONTOU (ver 2) CLUTHK (ver 2)
PLTTEM	L*4	-	Main (QUICK3)	TEMBRH (ver 2) CONTOU (ver 2)
PLTREF	L*4	-	Main (QUICK3)	TEMBRH (ver 2) REFLE1 (ver 2) CONTOU (ver 2)
PLTCLD	L*4	-	Main (QUICK3)	TEMBRH (ver 2) REFLE1 (ver 2) CONTOU (ver 2
ANG	L*4	-	Main (QUICK3)	REFLE1 (ver 2)

[1] For descriptions, see Section 4.2.1.4.

SKYLAB - Originated Common Trace

Main Program Package Name: QUICK

COMMON Name: VALUES

Variable[1]	Type	Dimension	Initialized or Reset in Subroutine	Referenced in Subroutine
R1	R*4	(35, 67)	REFLE1	REFLE1 REFLE2 CLUTHK
T1	R*4	(35, 67)	CLUTHK	DELTA1 CLUTHK WRITT
RR	R*4	(35, 67)	REFLE1 REFLE2 REFLE3	WRITT
Y1	R*4	(35, 67)	CLUTHK	REFLE2 REFLE3
X1	R*4	(35, 67)	REFLE2 REFLE3	REFLE2 REFLE3
D1	R*4	(35, 67)	REFLE2 REFLE3	REFLE2 REFLE3
E	R*4	(35, 67)	DELTA1	
E1	R*4	(35, 67)	DELTA1	WRITT
TBB	R*4	(35, 67)	TEMBRH	WRITT
R4	R*4	(35, 67)	REFLE3	REFLE3
LC	I*4	-	Main (QUICK)	REFLE1 CLUTHK WRITT
LR	I*4	-	Main (QUICK)	REFLE1 CLUTHK WRITT

[1] For descriptions, see Section 4.1.1.4.

SKYLAB - Originated Common Trace (Continued)

Variable[1]	Type	Dimension	Initialized or Reset in Subroutine	Referenced in Subroutine
LC2	I*4	-	Main (QUICK)	REFLE2 CLUTHK WRITT
LR2	I*4	-	Main (QUICK)	REFLE2 CLUTHK WRITT
LR3	I*4	-	Main (QUICK)	TEMBRH WRITT
LC3	I*4	-	Main (QUICK)	TEMBRH WRITT
LR4	I*4	-	Main (QUICK)	REFLE3 CLUTHK WRITT
LC4	I*4	-	Main (QUICK)	REFLE3 CLUTHK WRITT
Plot 1	L*4	-	Main (QUICK)	-
Plot 2	L*4	-	Main (QUICK)	-
Plot 3	L*4	-	Main (QUICK)	-
Plot 4	L*4	-	Main (QUICK)	-
Plot 5	L*4	-	Main (QUICK)	-

[1] For descriptions, see Section 4.1.1.4.

NonSKYLAB - Originated Common Trace

Main Program Package Name: Temperature Humidity Profile Program

COMMON Name: CHA

Variable[1]	Type	Dimension	Initialized or Reset in Subroutine	Referenced in Subroutine
RMSTMP (or Sum)	R*4	-	ESTLSQ	-
RMS	R*4	-	ESTH2O	ESTH2O

[1] For descriptions, see Section 3.1.3.4.

Main Program Package Name: Temperature Humidity Profile Program

COMMON Name: CO2

Variable[1]	Type	Dimension	Initialized or Reset in Subroutine	Referenced in Subroutine
TAUC (or TAUCO2) (or TAU)	R*4	(28, 100)	TRANSC	ESTH2O NCLOUD
CARBON (CO2 or CO, C1, C2 C3, C4, C5, C6, C7, C8)	R*4	(28, 9)	BLDATTHP	TRANSC
XK	R*4	(28)	BLDATTHP	TRNSIN (entry to TRANSC)
XKMULT	R*4	(28)	BLDATTHP	TRANSC

[1]For descriptions, see Section 3.1.3.4.

Main Program Package Name: Temperature Humidity Profile Program

COMMON Name: H2D

Variable[1]	Type	Dimension	Initialized or Reset in Subroutine	Referenced in Subroutine
TO	B*4	(99)	BLDATTHP	ESTLSQ NCLOND
QO	R*4	(99)	BLDATTHP ESTH2O	DTAUDU ESTH2O NCLOUD
TAU	R*4	(28,100)	DTANUDU ESTH2O NCLOUD TRANSW	ESTH2O ESTLSQ NCLOUD RAD RAD H2O
WATER (or CO, C1 C2, C3, C4, C5, C6, C7, C8)	R*4	(28,9)	BLDATTHP	DTAUDU TRANSW

[1] For descriptions, see Section 3.1.3.4.

Main Program Package Name: Temperature Humidity Profile Program

COMMON Name: SHLDT

Variable	Type	Dimension	Initialized or Reset in Subroutine	Referenced in Subroutine	Description
TSVAL	R*4	(100)	BLDATTHP	-[1]	Shelter temperature array for times of tape reading
TIMES	R*4	(100)	BLDATTHP	-[1]	Times associated with tape scans

NOTE: These were arbitrarly set to zero and never actually utilized since the data did not warrant further program development.

[1] Would have been referenced in GETSHL but this is presently a dummy routine.

Main Program Package Name: Temperature Humidity Profile Program

COMMON Name: UTOT

Variable	Type	Dimension	Initialized or Reset in Subroutine	Referenced in Subroutine	Description
U	R*4	-	TRANSW	-[1]	Total water vapor in atmosphere

[1] Referenced to other S00091 programs not suitable for SKYLAB Cloud studies conversion.

Main Program Package Name: Temperature Humidity Profile Program

COMMON Name: XXX

Variable[1]	Type	Dimension	Initialized or Reset in Subroutine	Referenced in Subroutine
TS	R*4	-	BLDATTHP	ESTH2O NCLOUD
PS	R*4	-	BLDATTHP	NCLOUD
PT	R*4	-	BLDATTHP	ESTH2O NCLOUD
DH	R*4	-	NCLOUD	ESTH2O NCLOUD
N	I*4	-	BLDATTHP	ESTH2O ESTLSQ NCLOUD
NL	I*4	-	BLDATTHP	ESTH2O ESTLSQ NCLOUD
R	R*4	-	BLDATTHP	ESTLSQ NCLOUD
MTOTAL	I*4	-	BLDATTHP	NCLOUD
P	R*4	99	NCLOUD	ESTH2O NCLOUD
T1	R*4	99	ESTLSQ	NCLOUD ESTH2O RADH2O } thru TRANSC } call TRANSW } sequences
HOTGRD	L*4	-	NCLOUD	ESTH2O NCLOUD

[1] For descriptions, see Section 3.1.3.4.

Main Program Package Name: Temperature Humidity Profile Program

COMMON Name: ZZZ

Variable[1]	Type	Dimension	Initialized or Reset in Subroutine	Referenced in Subroutine
XIO	R*4	(22)	RAD	ESTLSQ
PHI	R*4	(22,100)	ESTLSQ	ESTLSQ

[1]For descriptions, see Section 3.1.6.4.

APPENDIX M - TABLE OF IRRADIANCE VALUES USED AS BASIS FOR SUBTASK 1 INTERPOLATION ROUTINES

REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

SOLAR SPECTRAL IRRADIANCE - STANDARD CURVE

λ - Wavelength in micrometers

E_λ - Solar spectral irradiance averaged over small bandwidth centered at λ, in $W\ m^{-2}\ \mu m^{-1}$

$F_{o-\lambda}$ - Integrated solar irradiance in the wavelength range o to λ, in $W\ m^{-2}$

$D_{o-\lambda}$ - Percentage of solar constant associated with wavelengths shorter than λ

Solar constant - 1353 $W\ m^{-2}$

Note: lines indicate change in wavelength interval of integration

λ	E_λ	$E_{o-\lambda}$	$D_{o-\lambda}$	λ	E_λ	$E_{o-\lambda}$	$D_{o-\lambda}$	λ	E_λ	$E_{o-\lambda}$	$D_{o-\lambda}$
.115	.007	.0025	.0001	.510	1662	324.926	24.015	1.55	267	1146.109	87.665
.120	.900	.0048	.0003	.515	1633	334.214	24.701	1.60	245	1198.909	88.611
.125	.007	.0070	.0005	.520	1623	343.379	25.379	1.65	223	1210.609	89.475
.130	.007	.0071	.0005	.525	1652	352.591	26.059	1.70	202	1221.234	90.261
.140	.030	.0073	.0005	.530	1642	361.826	26.742	1.75	180	1230.784	90.967
.150	.070	.0078	.0005	.535	1618	370.976	27.418	1.80	159	1239.259	91.593
.160	.230	.0093	.0006	.540	1783	379.979	28.084	1.85	142	1246.784	92.149
.170	.630	.0136	.0010	.545	1754	388.821	28.737	1.90	126	1253.484	92.644
.180	1.250	.0230	.0016	.550	1725	397.519	29.360	1.95	114	1259.484	93.088
.190	2.710	.0428	.0031	.555	1720	406.131	30.017	2.00	103	1264.909	93.489
.200	10.7	.1098	.0081	.560	1695	414.669	30.648	2.1	90	1274.559	94.2024
.210	22.9	.2778	.0205	.565	1705	423.163	31.276	2.2	79	1283.009	94.8269
.220	57.5	.6796	.0502	.570	1712	431.711	31.907	2.3	69	1290.409	95.3739
.225	64.9	.9658	.0728	.575	1719	440.289	32.541	2.4	62	1296.959	95.8560
.230	66.7	1.3148	.0971	.580	1715	448.874	33.176	2.5	55	1302.809	96.2903
.235	59.3	1.6298	.1204	.585	1712	457.441	33.809	2.6	48	1307.959	96.6710
.240	63.0	1.9356	.1430	.590	1700	465.971	34.439	2.7	43	1312.509	97.0073
.245	72.3	2.2738	.1680	.595	1682	474.426	35.064	2.8	39	1316.609	97.3103
.250	70.4	2.6306	.1944	.600	1666	482.796	35.683	2.9	35	1320.309	97.5838
.255	104.0	3.0666	.2266	.605	1647	491.079	36.295	3.0	31	1323.609	97.8277
.260	130	3.6516	.269	.61	1635	499.284	36.902	3.1	26.0	1326.459	98.0383
.265	185	4.4391	.328	.62	1602	515.469	38.098	3.2	22.6	1328.889	98.2179
.270	232	5.4816	.405	.63	1570	531.329	39.270	3.3	19.2	1330.979	98.3724
.275	204	6.5716	.485	.64	1544	546.899	40.421	3.4	16.6	1332.769	98.5047
.280	222	7.6366	.564	.65	1511	562.174	41.550	3.5	14.6	1334.329	98.6200
.285	315	8.9791	.663	.66	1486	577.159	42.657	3.6	13.5	1335.734	98.7238
.290	482	10.9716	.810	.67	1456	591.869	43.744	3.7	12.3	1337.024	98.8192
.295	584	13.6366	1.007	.68	1427	606.264	44.810	3.8	11.1	1338.194	98.9056
.300	514	16.3016	1.210	.69	1402	620.429	45.855	3.9	10.3	1339.264	98.9847
.305	603	19.1741	1.427	.70	1364	634.264	46.879	4.0	9.5	1340.254	99.0579
.310	689	22.4041	1.655	.71	1344	647.849	47.882	4.1	8.70	1341.1641	99.12521
.315	764	26.0366	1.924	.72	1314	661.139	48.864	4.2	7.80	1341.9891	99.18618
.320	830	30.0216	2.218	.73	1290	674.159	49.826	4.3	7.10	1342.7341	99.24124
.325	975	34.5341	2.552	.74	1260	686.909	50.769	4.4	6.50	1343.4141	99.29150
.330	1059	39.6191	2.928	.75	1235	699.384	51.691	4.5	5.92	1344.0351	99.33740
.335	1081	44.9691	3.323	.76	1211	711.614	52.595	4.6	5.35	1344.5986	99.37905
.340	1074	50.3566	3.721	.77	1185	723.594	53.480	4.7	4.86	1345.1091	99.41678
.345	1069	55.7141	4.117	.78	1159	735.314	54.346	4.8	4.47	1345.5757	99.45127
.350	1093	61.1191	4.517	.79	1134	746.779	55.194	4.9	4.11	1346.0049	99.48299
.355	1083	66.5591	4.919	.80	1109	757.994	56.023	5.0	3.79	1346.3999	99.51219
.360	1068	71.9366	5.316	.81	1085	768.966	56.834	6	1.8200	1349.2049	99.71950
.365	1132	77.4366	5.723	.82	1060	779.694	57.627	7	.9900	1350.6099	99.82335
.370	1181	83.2191	6.150	.83	1036	790.174	58.401	8	.5850	1351.3974	99.88155
.375	1157	89.0641	6.582	.84	1013	800.419	59.158	9	.3670	1351.8734	99.91673
.380	1120	94.7566	7.003	.85	990	810.434	59.899	10	.2410	1352.1774	99.93920
.385	1098	100.3016	7.413	.86	968	820.224	60.622	11	.1650	1352.3834	99.95420
.390	1098	105.7916	7.819	.87	947	829.799	61.330	12	.1170	1352.5214	99.96462
.395	1189	111.5091	8.241	.88	926	839.164	62.022	13	.0851	1352.6224	99.97209
.400	1429	118.0541	8.725	.89	908	848.334	62.700	14	.0634	1352.6957	99.97758
.405	1644	125.7366	9.293	.90	891	857.329	63.365	15	.0481	1352.7524	99.98170
.410	1751	134.224	9.920	.91	880	866.184	64.019	16	.037100	1352.7950	99.98485
.415	1774	143.036	10.571	.92	869	874.929	64.665	17	.029100	1352.8281	99.98730
.420	1747	151.839	11.222	.93	858	883.564	65.304	18	.023100	1352.8542	99.98923
.425	1693	160.419	11.856	.94	847	892.089	65.934	19	.018601	1352.8751	99.99077
.430	1639	168.749	12.473	.95	837	900.509	66.556	20	.015200	1352.8922	99.99202
.435	1663	177.024	13.083	.96	820	908.794	67.168	25	.006170	1352.9454	99.99595
.440	1810	185.706	13.725	.97	803	916.909	67.768	30	.002970	1352.9693	99.99765
.445	1972	195.036	14.415	.98	786	924.849	68.355	35	.001600	1352.9797	99.99850
.450	2006	204.856	15.140	.99	767	932.609	68.928	40	.000942	1352.9850	99.99887
.455	2057	215.014	15.891	1.00	748	940.184	69.484	50	.000391	1352.9927	99.99946
.460	2066	225.321	16.653	1.05	668	975.594	72.105	60	.0001900	1352.9956	99.99967
.465	2048	235.606	17.413	1.10	593	1007.109	74.435	80	.0000616	1352.9981	99.99986
.470	2033	245.839	18.167	1.15	535	1035.309	76.519	100	.0000257	1352.9990	99.99992
.475	2044	256.001	18.921	1.20	485	1060.809	78.404	120	.0000126	1352.9994	99.99995
.480	2074	266.296	19.681	1.25	439	1083.484	80.109	150	.0000052	1352.9997	99.99997
.485	1976	276.421	20.433	1.30	397	1104.759	81.652	200	.0000016	1352.9998	99.99999
.490	1950	286.235	21.155	1.35	358	1123.634	83.047	250	.0000007	1352.9999	99.99999
.495	1960	296.011	21.878	1.40	337	1141.009	84.331	300	.0000003	1352.9999	99.99999
.500	1942	305.756	22.599	1.45	312	1157.234	85.530	400	.0000001	1352.9999	99.99999
.505	1920	315.421	23.312	1.50	288	1172.234	86.639	1000	.0000000	1353.0000	100.00000

SOLAR SPECTRAL IRRADIANCE
STANDARD CURVE, ABRIDGED VERSION

λ - WAVELENGTH IN μm

E_λ - SOLAR SPECTRAL IRRADIANCE AVERAGED OVER SMALL BANDWIDTH CENTERED AT λ, IN $W \cdot m^{-2} \cdot \mu m^{-1}$

$D_{o-\lambda}$ - PERCENTAGE OF THE SOLAR CONSTANT ASSOCIATED WITH WAVELENGTHS SHORTER THAN λ

SOLAR CONSTANT - 1353 $W \cdot m^{-2}$

λ	E_λ	$D_{o-\lambda}$	λ	E_λ	$D_{o-\lambda}$	λ	E_λ	$D_{o-\lambda}$
0.115	.007	1×10^{-4}	0.43	1639	12.47	0.90	891	63.37
0.14	.03	5×10^{-4}	0.44	1810	13.73	1.00	748	69.49
0.16	.23	6×10^{-4}	0.45	2006	15.14	1.2	485	78.40
0.18	1.25	1.6×10^{-3}	0.46	2066	16.65	1.4	337	84.33
0.20	10.7	8.1×10^{-3}	0.47	2033	18.17	1.6	245	88.61
0.22	57.5	0.05	0.48	2074	19.68	1.8	159	91.59
0.23	66.7	0.10	0.49	1950	21.15	2.0	103	93.49
0.24	63.0	0.14	0.50	1942	22.60	2.2	79	94.83
0.25	70.9	0.19	0.51	1882	24.01	2.4	62	95.86
0.26	130	0.27	0.52	1833	25.38	2.6	48	96.67
0.27	232	0.41	0.53	1842	26.74	2.8	39	97.31
0.28	222	0.56	0.54	1783	28.08	3.0	31	97.83
0.29	482	0.81	0.55	1725	29.38	3.2	22.6	98.22
0.30	514	1.21	0.56	1695	30.65	3.4	16.6	98.50
0.31	689	1.66	0.57	1712	31.91	3.6	13.5	98.72
0.32	830	2.22	0.58	1715	33.18	3.8	11.1	98.91
0.33	1059	2.93	0.59	1700	34.44	4.0	9.5	99.06
0.34	1074	3.72	0.60	1666	35.68	4.5	5.9	99.34
0.35	1093	4.52	0.62	1602	38.10	5.0	3.8	99.51
0.36	1068	5.32	0.64	1544	40.42	6.0	1.8	99.72
0.37	1181	6.15	0.66	1486	42.66	7.0	1.0	99.82
0.38	1120	7.00	0.68	1427	44.81	8.0	.59	99.88
0.39	1098	7.82	0.70	1369	46.88	10.0	.24	99.94
0.40	1429	8.73	0.72	1314	48.86	15.0	4.8×10^{-2}	99.98
0.41	1751	9.92	0.75	1235	51.69	20.0	1.5×10^{-2}	99.99
0.42	1747	11.22	0.80	1109	56.02	50.0	3.9×10^{-4}	100.00

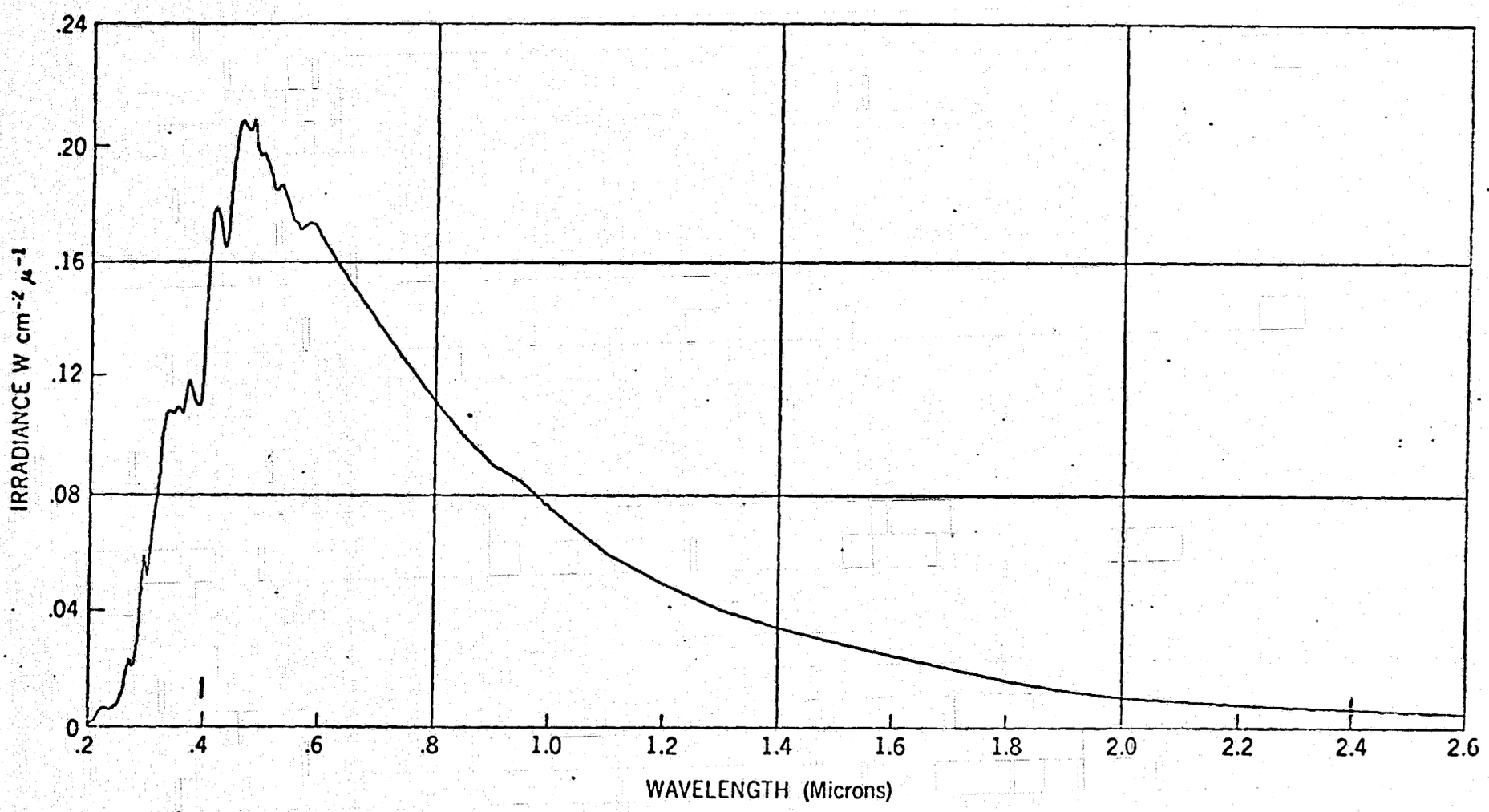
REPRODUCIBILITY OF THE
ORIGINAL PAGE IS POOR

SOLAR SPECTRAL IRRADIANCE: 3000 TO 61000: 1 A INTERVALS

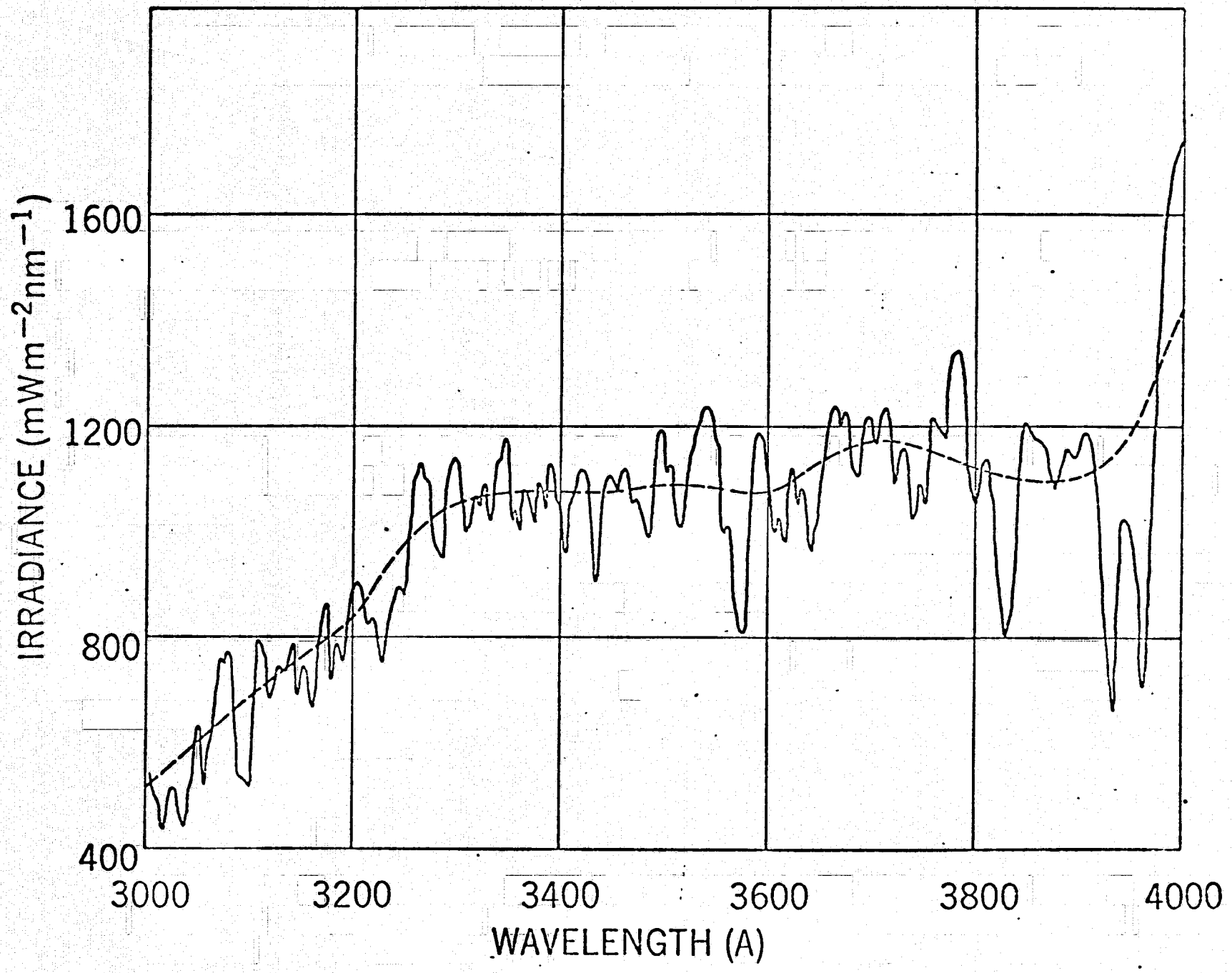
WAVELENGTH IN A; IRRADIANCE IN MW PER M SQ A=LA SM W=MW

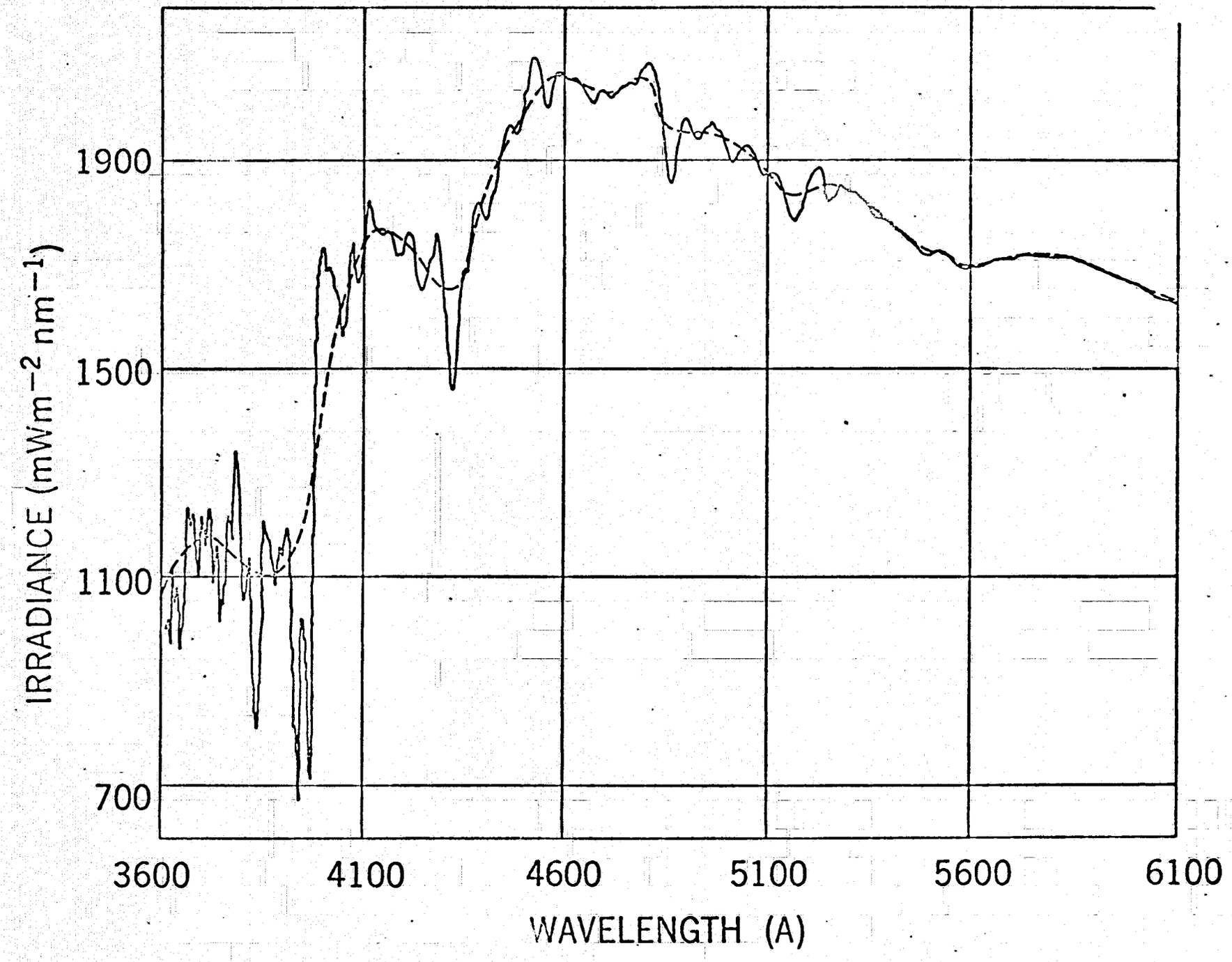
	0	1	2	3	4	5	6	7	8	9
3000	570	564	557	564	562	515	527	522	514	510
3010	511	512	492	477	493	481	482	487	481	490
3020	513	527	535	540	541	541	540	535	523	504
3030	491	484	477	478	480	485	488	477	494	523
3040	534	544	545	544	562	544	611	630	652	653
3050	647	636	591	567	547	544	539	545	614	617
3060	623	628	640	658	680	704	720	735	748	771
3070	777	775	772	770	772	778	787	791	791	792
3080	787	776	732	691	646	609	575	564	561	560
3090	560	559	556	550	553	549	545	543	542	544
3100	553	644	674	743	765	799	818	818	805	796
3110	791	785	784	775	760	722	709	706	707	712
3120	719	727	735	744	751	757	762	763	762	760
3130	757	754	751	751	754	761	771	780	788	793
3140	804	800	727	713	711	713	719	730	742	752
3150	761	768	765	764	758	745	705	694	689	687
3160	688	697	712	730	734	794	818	822	834	837
3170	835	828	822	819	814	744	734	734	741	750
3180	774	793	801	803	801	795	783	772	772	775
3190	784	800	814	830	843	854	863	864	865	861
3200	915	918	918	918	915	903	892	880	868	864
3210	850	844	843	845	848	850	851	850	847	835
3220	825	803	786	777	771	771	770	782	796	808
3230	817	824	834	842	854	866	873	884	893	899
3240	904	908	912	911	908	900	894	902	915	927
3250	962	971	995	1022	1047	1066	1102	1111	1112	1110
3260	1129	1135	1139	1139	1122	1115	1112	1111	1110	1108
3270	1102	1079	1068	1043	1015	994	987	982	979	975
3280	971	967	963	964	969	1003	1021	1066	1109	1112
3290	1119	1127	1140	1147	1150	1151	1149	1142	1135	1118
3300	1103	1086	1061	1034	1016	1011	1013	1020	1022	1021
3310	1023	1030	1041	1054	1065	1073	1077	1076	1062	1060
3320	1065	1078	1086	1091	1095	1103	1076	1049	1037	1031
3330	1035	1040	1046	1066	1080	1105	1137	1145	1145	1140
3340	1151	1161	1175	1182	1184	1185	1183	1155	1120	1074
3350	1039	1033	1037	1045	1050	1030	1021	1017	1020	1042
3360	1074	1086	1089	1090	1087	1079	1075	1055	1044	1041
3370	1036	1031	1028	1031	1052	1095	1103	1115	1114	1100
3380	1065	1059	1055	1066	1094	1125	1135	1137	1136	1131
3390	1110	1098	1079	1072	1075	1067	1044	1033	1018	987
3400	960	974	973	975	981	990	1015	1038	1064	1073
3410	1076	1080	1088	1095	1104	1114	1122	1126	1126	1127
3420	1125	1120	1104	1091	1079	1066	1036	1023	996	956
3430	921	923	927	933	941	969	1054	1073	1063	1045
3440	1103	1110	1115	1114	1123	1121	1115	1117	1112	1103
3450	1096	1091	1091	1096	1100	1108	1115	1122	1124	1131
3460	1124	1124	1111	1093	1026	1069	1067	1066	1066	1072
3470	1073	1073	1071	1068	1062	1055	1047	1038	1028	1018
3480	1010	1006	1004	1004	1012	1027	1037	1050	1046	1042
3490	1169	1182	1201	1208	1208	1207	1201	1182	1128	1127
3500	1127	1131	1136	1138	1139	1126	1103	1075	1044	1032
3510	1024	1022	1021	1023	1027	1034	1059	1060	1060	1062
3520	1116	1126	1140	1155	1154	1163	1171	1186	1200	1206
3530	1213	1220	1225	1231	1237	1241	1246	1246	1247	1246
3540	1248	1246	1241	1236	1233	1215	1191	1183	1181	1184
3550	1187	1197	1204	1217	1213	1212	1219	1224	1026	1026
3560	1024	1007	981	967	949	891	872	856	843	833
3570	830	828	827	827	828	829	841	855	844	820
3580	988	1050	1074	1122	1150	1178	1191	1195	1197	1196
3590	1196	1192	1188	1182	1180	1134	1114	1128	1088	1080
3600	1047	1035	1021	1015	1016	1022	1031	1039	1039	1035
3610	1017	1037	1051	967	990	994	1031	1079	1107	1127
3620	1131	1131	1116	1103	1075	1068	1070	1074	1065	1059
3630	1045	1043	1041	1037	1038	1018	993	988	981	977
3640	982	1005	1018	1031	1027	1044	1057	1060	1059	1113
3650	1127	1139	1152	1172	1189	1208	1213	1221	1233	1241
3660	1245	1247	1246	1248	1243	1227	1214	1218	1220	1227
3670	1234	1237	1239	1234	1223	1207	1191	1159	1149	1134
3680	1120	1124	1121	1120	1123	1133	1131	1144	1171	1194
3690	1210	1221	1227	1226	1233	1224	1224	1213	1195	1183
3700	1160	1174	1181	1186	1196	1211	1226	1232	1237	1240
3710	1243	1245	1244	1233	1222	1211	1179	1167	1167	1169
3720	1173	1178	1179	1183	1194	1198	1196	1191	1184	1179
3730	1151	1163	1174	1183	1190	1196	1203	1207	1205	1208
3740	1208	1207	1199	1190	1196	1197	1213	1227	1226	1228
3750	1225	1222	1218	1213	1207	1204	1194	1193	1196	1196
3760	1249	1263	1311	1319	1327	1333	1347	1344	1351	1353
3770	1355	1355	1351	1342	1336	1312	1294	1290	1286	1283
3780	1122	1107	1096	1087	1080	1075	1072	1078	1089	1078
3790	1077	1090	1104	1113	1133	1144	1144	1150	1151	1149
3800	1146	1144	1140	1135	1134	1134	1134	1134	1134	1134
3810	1134	1134	1134	1134	1134	1134	1134	1134	1134	1134
3820	1134	1134	1134	1134	1134	1134	1134	1134	1134	1134
3830	1134	1134	1134	1134	1134	1134	1134	1134	1134	1134
3840	1134	1134	1134	1134	1134	1134	1134	1134	1134	1134
3850	1134	1134	1134	1134	1134	1134	1134	1134	1134	1134
3860	1134	1134	1134	1134	1134	1134	1134	1134	1134	1134
3870	1134	1134	1134	1134	1134	1134	1134	1134	1134	1134
3880	1134	1134	1134	1134	1134	1134	1134	1134	1134	1134
3890	1134	1134	1134	1134	1134	1134	1134	1134	1134	1134
3900	1186	1192	1199	1202	1203	1202	1199	1196	1194	1196
3910	1196	1195	1194	1179	1032	1012	978	961	955	956
3920	849	832	816	799	776	756	731	706	693	687
3930	686	721	800	832	849	871	891	1032	1036	1038
3940	1039	1038	1038	1038	1032	1026	1022	1016	1009	988
3950	970	958	944	923	911	871	757	736	730	731
3960	734	734	693	693	693	691	1017	1043	1041	1022
3970	1035	1031	1031	1045	1054	1067	1076	1079	1076	1075
3980	1059	1057	1042	1043	1044	1044	1044	1044	1044	1044
3990	1783	1718	1717	1726	1736	1734	1734	1742	1744	1740
4000	1739	1734	1740	1740	1724	1715	1707	1703	1701	1709
4010	1699	1693	1701	1703	1705	1707	1704	1704	1704	1704
4020	1707	1706	1704	1700	1691	1691	1689	1682	1674	1674
4030	1657	1657	1656	1655	1654	1652	1653	1644	1645	1633
4040	1617	1604	1599	1591	1587	1582	1578	1576	1576	1579
4050	1582	1586	1584	1584	1591	1592	1592	1597	1594	1604
4060	1612	1623	1644	1652	1665	1677	1705	1729	1733	1736
4070	1760	1763	1766	1768	1750	1752	1749	1742	1727	1709
4080	1704	1692	1684	1674	1675	1676	1677	1674	1681	1683
4090	1686	1688	1691	1698	1701	1707	1713	1720	1725	1732
4100	1739	1745	1750	1754	1759	1766	1771	1782	1801	1807
4110	1814	1821	1825	1827	1824	1828	1827	1825	1823	1821
4120	1818	1815	1810	1805	1799	1793	1791	1787	1784	1782
4130	1783	1779	1776	1774	1777	1777	1776	1775	1774	1773
4140	1773	1772	1771	1770	1769	1769	1768	1768	1768	1768
4150	1769	1769	1770	1771	1771	1772	1773	1774	1774	1773
4160	1775	1775	1776	1776	1776	1776	1775	1775	1774	1773
4170	1772	1772	1770	1769	1767	1765	1762	1757	1754	1749
4180	1742	1735	1730	1727	1727	1727	1728	1729	1731	1732
4190	1732	1732	1732	1731	1730	1730	1729	1729	1729	1730
4200	1731	1733	1736	1740	1744	1748	1751	1755	1759	1761
4210	1764	1766	1771	1772	1772	1771	1770	1769	1768	1767
4220	1766	1764	1763	1760	1757	1754	1751	1749	1747	1745
4230	1731	1702	1690	1690	1684	1679	1675	1672	1669	1667
4240	1665	1663	1662	1661	1660	1660	1661	1661	1663	1665
4250	1669	1671	1673	1675	1677	1679	1681	1683	1685	1687
4260	1693	1692	1695	1698	1702	1706	1710	1714	1718	1722
4270	1725	1727	1726	1730	1730	1731	1729	1735	1741	1740
4280	1753	1757	1761	1765	1767	1766	1760	1754	1742	1733
4290	1732	1730	1723	1717	1712	1706	1699	1690	1680	1672
4300	1666	1657	1646	1636	1631	1621	1609	1597	1588	1580
4310	1564	1553	1544	1529	1518	1508	1499	1490	1482	1476
4320	1472	1471	1477	1477	1479	1480	1484	1504	1505	1518
4330	1526	1532	1544	1554	1563	1565	1595	1600	1606	1620
4340	1642	1653	1659	1663	1666	1662	1665	1668	1672	1679
4350	1699	1702	1700	1699	1698	1697	1696	1695	1693	1692
4360	1692	1692	1692	1693	1706	1719	1725	1731	1743	1754
4370	1769	1768	1780	1786	1793	1798	1802	1803	1805	1807
4380	1809	1812	1815	1816	1819	1821	1822	1823	1823	1822
4390	1821	1821	1820	1819	1818	1817	1817	1816	1814	1812
4400	1810	1807	1804	1801	1797	1794	1791	1790	1789	1789
4410	1790	1790	1791	1792	1794	1795	1799	1803	1810	1818
4420	1823	1827	1831	1834	1837	1840	1840	1841	1842	1844
4430	1846	1848	1850	1853	1857	1856	1859	1875	1881	1883
4440	1890	1896	1903	1908	1913	1917	1919	1921	1922	1924
4450	1927	1931	1937	1941	1947	1951	1954	1958	1961	1964
4460	1967	1969	1970	1971	1973	1974	1975	1975	1976	1976
4470	1977	1977	1976	1976	1974	1972	1969	1966	1967	1966
4480	1965	1963	1962	1961	1960	1960	1960	1961	1962	1962
4490	1963	1965	1966	1968	1971	1974	1976	1977	1981	1983
4500	1990	1994	1998	2003	2010	2015	2022	2030	2037	2039
4510	2040	2040	2040	2041	2041	2041	2044	2056	2063	2070
4520	2076	2082	2088	2092	2097	2099	2101	2103	2103	2103
4530	2103	2103	2103	2102	2102	2101	2101	2100	2099	2097
4540	2095	2093	2091	2088	2086	2085	2078	2077	2064	2060
4550	2061	2056	2050	2041	2035	2028	2022	2018	2013	2010
4560	2011	2011	2012	2012	2013	2013	2013	2013	2013	2013
4570	2014	2015	2017	2021	2026	2033	2034	2044	2050	2052
4580	2054	2052	2055	2057	2060	2071	2073	2074	2076	2076
4590	2077	2077	2077	2077	2077	2077	2076	2076	2075	2074
4600	2074	2073	2073	2072	2071	2070	2069	2068	2067	2066
4610	2065	2065	2064	2063	2062	2061	2060	2060	2059	2059
4620	2060	2060	2060	2060	2060	2060	2062	2062	2062	2063
4630	2063	2063	2063	2063	2063	2062	2062	2062	2062	2061
4640	2061	2061	2060	2059	2058	2057	2056	2055	2055	2054
4650	2054	2053	2052	2050	2049	2048	2044	2043	2040	2038
4660	2037	2035	2034	2033	2032	2032	2031	2030	2029	2028
4670	2027	2026	2025	2023	2021	2019	2017	2017	2017	2018
4680	2019	2021	2022	2024	2025	2026	2028	2030	2032	2034
4690	2035	2037	2038	2039	2041	2041	2042	2043	2043	2043
4700	2044	2044	2044	2044	2043	2043	2042	2041	2040	2039
4710	2039	2038	2037	2035	2034	2032	2031	2030	2029	2029
4720	2028	2028	2027	2027	2026	2027	2027	2029	2029	2030
4730	2031	2032	2034	2035	2035	2036	2037	2037	2038	2038
4740	2039	2038	2041	2042	2043	2044	2045	2046	2047	2047
4750	2046	2048	2048	2048	2047	2047	2047	2047	2047	2047
4760	2047	2048	2048	2049	2050	2050	2051	2052	2053	2054
4770	2054	2055	2055	2055	2055	2056	2055	2054	2052	2051
4780	2050	2050	2051	2051	2053	2054	2055	2057	2059	2061
4790	2062	2063	2065	2067	2069	2070	2072	2073	2075	2076
4800	2076	2078	2079	2080	2082	2083	2084	2085	2086	2086
4810	2086	2089	2090	2091	2091	2092	2092	2092	2092	2091
4820	2090	2088	2087	2085	2084	2083	2082	2081	2080	2079
4830	2079	2078	2076	2076	2073	2071	2069	2067	2067	2059
4840	2063	2061	2059	2057	2032	2028	2019	2010	2006	1999
4850	1990	1979	1969	1958	1950	1943	1937	1932	1923	1915
4860	1908	1902	1896	1887	1877	1877	1873	1870	1868	1866
4870	1864	1862	1861	1860	1859	1859	1860	1861	1864	1867
4880	1875	1881	1888	1892	1896	1901	1905	1910	1926	1933
4890	1940	1947	1951	1955	1960	1964	1968	1970	1972	1974
4900	1976	1981	1992	1984	1985	1986	1987	1987	1987	1987
4910	1987	1987	1985	1984	1983	1981	1981	1981	1979	1975
4920	1974	1972	1971	1969	1967	1965	1964	1962	1962	1957
4930	1953	1951	1949	1947	1945	1944	1943	1942	1940	1939
4940	1943	1943	1944	1945	1946	1946	1953	1952	1954	1950
4950	1950	1960	1962	1963	1964	1965	1966	1967	1967	1969
4960	1968	1969	1969	1969	1970	1970	1971	1972	1973	1974
4970	1975	1977	1979	1980	1981	1980	1980	1983	1983	1982
4980	1977	1975	1972	1969	1968	1966	1963	1963	1962	1960
4990	1962	1962	1961	1961	1959	1956	1954	1951	1949	1947
5000	1947	1944	1941	1939	1937	1935	1934	1931	1929	1927
5010	1926	1922	1920	1918	1915	1912	1909	1908	1906	1904
5020	1901	1900	1899	1898	1897	1897	1896	1896	1895	1895
5030	1894	1893	1893	1893	1893	1893	1893	1893	1893	1893
5040	1893	1893	1893	1893	1893	1893	1893	1893	1893	1893
5050	1893	1893	1893	1893	1893	1893	1893	1893	1893	1893
5060	1893	1893	1893	1893	1893	1893	1893	1893	1893	1893
5070	1893	1893	1893	1893	1893	1893	1893	1893	1893	1893
5080	1893	1893	1893	1893	1893	1893	1893	1893	1893	1893
5090	1893	1893	1893	1893	1893	1893	1893	1893	1893	1893
5100	1870	1869	1869	1869	1869	1869	1869	1869	1870	1870
5110	1871	1872	1873	1875	1876	1877	1877	1876	1876	1876
5120	1876	1876	1876	1876	1876	1876	1876	1876	1876	1876
5130	1876	1876	1876	1876	1876	1876	1876	1876	1876	1876
5140	1859	1857	1855	1854	1852	1850	1847	1844	1844	1844
5150	1838	1831	1829	1825	1824	1822	1822	1817	1816	1816
5160	1805	1803	1801	1799	1797	1795	1794	1792	1792	1793
5170	1788	1786	1785	1784	1783	1781	1782	1782	1783	1784
5180	1788	1786	1785	1784	1784	1783	1782	1782	1782	1782
5190	1802	1803	1805	1807	1810	1813	1818	1822	1826	1828
5200	1837	1840	1843	1846	1849	1852	1855	1856	1858	1860
5210	1862	1865	1867	1869	1871	1874	1875	1877	1878	1879
5220	1879	1879	1880	1881	1882	1881	1882	1883	1885	1888
5230	1887	1887	1886	1884	1884	1884	1883	1883	1882	1882
5240	1884	1884	1885	1887	1888	1888	1890	1890	1890	1890
5250	1891	1893	1893	1894	1894	1894	1894	1894	1894	1894
5260	1921	1920	1920	1921	1921	1922	1924	1925	1925	1926
5270	1926	1926	1926	1927	1928	1930	1932	1933	1933	1934
5280	1936	1938	1939	1940	1942	1944	1946	1948	1950	1952
5290	1955	1955	1955	1955	1954	1953	1952	1951	1950	1949
5300	1947	1946	1945	1944	1944	1943	1942	1941	1940	1940
5310	1939	1938	1938	1937	1937	1937	1935	1936	1936	1938
5320	1938	1938	1937	1938	1938	1938	1939	1940	1939	1938
5330	1938	1938	1938	1938	1938	1938	1938	1938	1938	1938
5340	1938	1929	1924	1924	1924	1924	1927	1925	1923	1922
5350	1921	1920	1919	1918	1917	1916	1915	1914	1913	1911
5360	1909	1907	1905	1903	1901	1749	1749	1747	1746	1745
5370	1743	1742	1740	1740	1740	1740	1740	1740	1740	1740
5380	1739	1740	1740	1740	1740	1740	1740	1740	1740	1740
5390	1787	1787	1787	1787	1787	1787	1787	1787	1787	1787
5400	1786	1785	1785	1785	1785	1784	1783	1782	1782	1781
5410	1780	1780	1779	1778	1778	1777	1776	1775	1774	1774
5420	1773	1771	1770	1769	1768	1767	1767	1767	1767	1766
5430	1766	1766	1766	1765	1765	1765	1765	1764	1764	1763
5440	1763	1762	1761	1761	1761	1760	1759	1758	1757	1757
5450	1755	1754	1753	1752	1751	1750	1750	1749	1748	1748
5460	1747	1747	1747	1746	1746	1745	1744	1744	1743	1742
5470	1741	1741	1740	1739	1739	1738	1738	1735	1735	1734
5480	1732	1731	1731	1730	1729	1729	1727	1726	1725	1724
5490	1723	1722	1721	1721	1720	1720	1720	1719	1719	1719
5500	1719	1719	1719	1719	1720	1720	1721	1721	1722	1722
5510	1723	1723	1723	1724	1724	1725	1726	1726	1727	1728
5520	1729	1729	1730	1731	1731	1732	1732	1732	1732	1731
5530	1731	1731	1731	1731	1730	1730	1730	1730	1729	1729
5540	1728	1727	1727	1727	1727	1726	1725	1725	1725	1724
5550	1722	1722	1721	1720	1719	1719	1719	1718	1717	1716
5560	1714	1713	1712	1711	1710	1709	1709	1708	1707	1706
5570	1704	1703	1702	1701	1701	1700	1699	1698	1697	1697
5580	1696	1696	1695	1695	1694	1694	1693	1693	1693	1693
5590	1692	1692	1692	1692	1692	1692	1692	1692	1692	1692
5600	1693	1693	1693	1694	1694	1695	1695	1695	1695	1695
5610	1696	1696	1696	1696	1697	1697	1697	1697	1697	1698
5620	1698	1698	1698	1698	1699	1699	1699	1699	1699	1699
5630	1699	1699	1699	1699	1700	1700	1700	1700	1701	1701
5640	1701	1701	1702	1702	1702	1702	1703	1703	1703	1704
5650	1704	1705	1705	1705	1705	1705	1706	1706	1707	1707
5660	1708	1708	1709	1709	1710	1710	1711	1711	1711	1712
5670	1712	1712	1712	1712	1712	1713	1713	1713	1713	1713
5680	1713	1713	1713	1713	1713	1713	1713	1713	1713	1713
5690	1711	1711	1711	1711	1711	1711	1710	1710	1710	1710
5700	1710	1710	1710	1711	1711	1711	1711	1711	1711	1711
5710	1712	1712	1712	1712	1712	1712	1713	1713	1713	1713
5720	1714	1714	1715	1716	1716	1717	1716	1716	1717	1717
5730	1718	1718	1718	1719	1719	1719	1719	1719	1719	1719
5740	1720	1720	1720	1720	1720	1720	1720	1720	1720	1720
5750	1719	1719	1719	1719	1719	1719	1719	1719	1719	1719
5760	1719	1719	1719	1719	1719	1719	1719	1719	1718	1718
5770	1718	1718	1718	1718	1718	1718	1718	1717	1717	1717
5780	1717	1717	1717	1717	1717	1717	1717	1716	1716	1716
5790	1716	1716	1716	1715	1715	1715	1715	1714	1714	1714
5800	1714	1714	1714	1714	1714	1714	1714	1713	1713	1713
5810	1713	1713	1713	1713	1714	1714	1714	1714	1714	1714
5820	1714	1714	1714	1714	1714	1714	1714	1713	1713

SOLAR SPECTRAL IRRADIANCE (NASA STANDARD)



M-5


IRRADIANCE (mWm^{-2}nm^{-1})
1600
1200
800
400
3000
3200
3400
3600
3800
4000
WAVELENGTH (A)


1900
1500
1100
700
IRRADIANCE (mWm^{-2} nm^{-1})
3600
4100
4600
5100
5600
6100
WAVELENGTH (A)

APPENDIX N - CONTENTS OF S-192 OUTPUT ARRAY

Integer* 2 I*2 (16 bits, with 2 bytes, each)	Integer* 4 I*4 (32 bits, with 4 bytes, each)	Bytes	Description
1		1-2	Channel number
2		3-4	Total number of active channels
3		5-6	Number of video elements/scans/ channels
4		7-8	Line number
5		9-10	Total number of lines
6		11-12	Month
7		13-14	Day
8		15-16	Year
9		17-18	Hour
10		19-20	Minute
11		21-22	Second
12		23-24	Tenths of milliseconds
	7	25-28	Tenths of milliseconds into day
15-36		29-72	Channel status 0 active 1 not active
37		73-74	Degree } Latitude
38		75-76	Minute } Latitude
39		77-78	Quarter-second } Latitude
40		79-80	Degree } Longitude (w+)
41		81-82	Minute } Longitude (w+)
42		83-84	Quarter-second } Longitude (w+)

Integer* 2 I*2 (16 bits, with 2 bytes, each)	Integer*4 I*4 (32 bits, with 4 bytes, each)	Bytes	Description	
43		85-86	Degree	Roll
44		87-88	Minute	
45		89-90	Quarter-second	
46		91-92	Degree	Pitch
47		93-94	Minute	
48		95-96	Quarter-second	
49		97-98	Degree	Yaw
50		99-100	Minute	
51		101-102	Quarter-second	
52		103-104	Degree	Zenith Angle
53		105-106	Minute	
54		107-108	Quarter-second	
55-57		109-114	Degree Minute Quarter-second Latitude	Center of 1st pixel
58-60		115-120	Degree Minute Quarter-second W. Longitude	

Integer* 2 I*2 (16 bits, with 2 bytes, each)	Integer* 4 I*4 (32 bits, with 4 bytes, each)	Bytes	Description
61-63		121-126	Degree Minute Quarter-second Latitude — Center of center pixel
64-66		127-132	Degree Minute Quarter-second W. Longitude — Center of center pixel
67-69		133-138	Degree Minute Quarter-second Latitude — Center of last pixel
70-72		139-144	Degree Minute Quarter-second W. Longitude — Center of last pixel
	37	145-148	Hot blackbody temperature sensor '1A' (tenth deg Kelvin)
	38	149-152	Cold blackbody temperature sensor '2A' (tenth deg K)
	39	153-156	Lamp 1 Current Motor (milliamps)
	40	157-160	Redundant blackbody temperature sensor '3A' (tenth deg K)

Integer* 2 I*2 (16 bits, with 2 bytes, each)	Integer* 4 I*4 (32 bits, with 4 bytes, each)	Bytes	Description
70-72 (cont'd)	41	161-164	Lamp 2 current motor (milliamps)
	42	165-168	Detector array temperature sensor 'A' (tenth deg K)
	43	169-172	Primary mirror temperature sensor (tenth deg K)
	44	173-176	Secondary mirror temperature sensor (tenth deg K)
	45	177-180	Primary mirror temperature sensor 2 (tenth deg K)
	46	181-184	Aspheric mirror temperature sensor (tenth deg K)
	47	185-188	Cover temperature sensor (tenth deg K)
	48	189-192	Scan motor temperature sensor (tenth deg K)
	49	193-196	Cooler motor temperature (tenth deg K)
	50	197-200	Scanner electronic temperature sensor (tenth deg K)
	51	201-204	Monochromator temperature (tenth deg K)

Integer* 2 I*2 (16 bits, with 2 bytes, each)	Integer* 4 I*4 (32 bits, with 4 bytes, each)	Bytes	Description
70-72 (cont'd)	52	205-208	Digital electronic temperature (tenth deg K)
	53	209-212	Detector array temperature sensor 'B' (tenth deg K)
	54	213-216	Hot blackbody temperature sensor '2B' (tenth deg K)
	55	217-220	Cold blackbody temperature sensor '2B' (tenth deg K)
	56	221-224	Redundant blackbody temperature sensor '3B' (tenth deg K)
	57	225-228	Scan line offset time (tenth microsecond)
	58	229-232	Scan shift delay flag
	59	233-236	Azimuth heading (degree)
	60	237-240	Ground velocity (meters/sec)
	61	241-244	Altitude (meters)
	62	245-248	Orbital inclination angle quarter arc sec
	63	249	Pixel radiance values for line - open ended depending on number of pixels on line

Integer* 2 I*2 (16 bits, with 2 bytes, each)	Integer* 4 I*4 (32 bits, with 4 bytes, each)	Bytes	Description
70-72 (cont'd)	63	249	Pixel radiance values for line - open ended depending on number of pixels on line

NOTE: Integer* 4 words 37 thru 57 are alphanumeric (A4) printout.

APPENDIX O - MODEL ATMOSPHERES USED AS A BASIS OF COMPUTATION OF ATMOSPHERIC OPTICAL PROPERTIES

MIDLATITUDE SUMMER					
Ht. (km)	Pressure (mb)	Temp. (°K)	Density (g/m^3)	Water Vapor (g/m^3)	Ozone (g/m^3)
0	1.013E+03	294.0	1.191E+03	1.4E+01	6.0E-05
1	9.020E+02	290.0	1.080E+03	9.3E+00	6.0E-05
2	8.020E+02	285.0	9.757E+02	5.9E+00	6.0E-05
3	7.100E+02	279.0	8.846E+02	3.3E+00	6.2E-05
4	6.280E+02	273.0	7.998E+02	1.9E+00	6.4E-05
5	5.540E+02	267.0	7.211E+02	1.0E+00	6.6E-05
6	4.870E+02	261.0	6.487E+02	6.1E-01	6.9E-05
7	4.260E+02	255.0	5.830E+02	3.7E-01	7.5E-05
8	3.720E+02	248.0	5.225E+02	2.1E-01	7.9E-05
9	3.240E+02	242.0	4.669E+02	1.2E-01	8.6E-05
10	2.810E+02	235.0	4.159E+02	6.4E-02	9.0E-05
11	2.430E+02	229.0	3.693E+02	2.2E-02	1.1E-04
12	2.090E+02	222.0	3.269E+02	6.0E-03	1.2E-04
13	1.790E+02	216.0	2.882E+02	1.8E-03	1.5E-04
14	1.530E+02	216.0	2.464E+02	1.0E-03	1.8E-04
15	1.300E+02	216.0	2.104E+02	7.6E-04	1.9E-04
16	1.110E+02	216.0	1.797E+02	6.4E-04	2.1E-04
17	9.500E+01	216.0	1.535E+02	5.6E-04	2.4E-04
18	8.120E+01	216.0	1.305E+02	5.0E-04	2.8E-04
19	6.950E+01	217.0	1.110E+02	4.9E-04	3.2E-04
20	5.950E+01	218.0	9.453E+01	4.5E-04	3.4E-04
21	5.100E+01	219.0	8.056E+01	5.1E-04	3.6E-04
22	4.370E+01	220.0	6.872E+01	5.1E-04	3.6E-04
23	3.760E+01	222.0	5.867E+01	5.4E-04	3.4E-04
24	3.220E+01	223.0	5.014E+01	6.0E-04	3.2E-04
25	2.770E+01	224.0	4.288E+01	6.7E-04	3.0E-04
30	1.320E+01	234.0	1.322E+01	3.6E-04	2.0E-04
35	6.520E+00	245.0	6.519E+00	1.1E-04	9.2E-05
40	3.330E+00	258.0	3.330E+00	4.3E-05	4.1E-05
45	1.760E+00	270.0	1.757E+00	1.9E-05	1.3E-05
50	9.510E-01	276.0	9.512E-01	6.3E-06	4.3E-06
70	6.710E-02	218.0	6.706E-02	1.4E-07	8.6E-08
100	3.000E-04	210.0	5.000E-04	1.0E-09	4.3E-11

MIDLATITUDE WINTER					
Ht. (km)	Pressure (mb)	Temp. (°K)	Density (g/m^3)	Water Vapor (g/m^3)	Ozone (g/m^3)
0	1.018E+03	272.2	1.301E+03	3.5E+00	6.0E-05
1	8.973E+02	268.7	1.162E+03	2.5E+00	5.4E-05
2	7.897E+02	265.2	1.037E+03	1.8E+00	4.9E-05
3	6.938E+02	261.7	9.230E+02	1.2E+00	4.9E-05
4	6.081E+02	255.7	8.282E+02	6.6E-01	4.9E-05
5	5.313E+02	249.7	7.411E+02	3.8E-01	5.8E-05
6	4.627E+02	243.7	6.614E+02	2.1E-01	6.4E-05
7	4.016E+02	237.7	5.886E+02	8.5E-02	7.7E-05
8	3.473E+02	231.7	5.222E+02	3.5E-02	9.0E-05
9	2.992E+02	225.7	4.619E+02	1.6E-02	1.2E-04
10	2.568E+02	219.7	4.072E+02	7.5E-03	1.6E-04
11	2.199E+02	219.2	3.496E+02	6.9E-03	2.1E-04
12	1.882E+02	218.7	2.999E+02	6.0E-03	2.6E-04
13	1.610E+02	218.2	2.572E+02	1.8E-03	3.0E-04
14	1.378E+02	217.7	2.206E+02	1.0E-03	3.2E-04
15	1.178E+02	217.2	1.890E+02	7.6E-04	3.4E-04
16	1.007E+02	216.7	1.620E+02	6.4E-04	3.6E-04
17	8.610E+01	216.2	1.388E+02	5.6E-04	3.9E-04
18	7.350E+01	215.7	1.188E+02	5.0E-04	4.1E-04
19	6.280E+01	215.2	1.017E+02	4.9E-04	4.3E-04
20	5.370E+01	215.2	8.690E+01	4.5E-04	4.5E-04
21	4.580E+01	215.2	7.421E+01	5.1E-04	4.3E-04
22	3.910E+01	215.2	6.338E+01	5.1E-04	4.3E-04
23	3.340E+01	215.2	5.415E+01	5.4E-04	3.9E-04
24	2.860E+01	215.2	4.624E+01	6.0E-04	3.6E-04
25	2.430E+01	215.2	3.950E+01	6.7E-04	3.4E-04
30	1.110E+01	217.4	1.783E+01	3.6E-04	1.9E-04
35	5.180E+00	227.8	7.924E+00	1.1E-04	9.2E-05
40	2.530E+00	243.2	3.625E+00	4.3E-05	4.1E-05
45	1.290E+00	258.5	1.741E+00	1.9E-05	1.3E-05
50	6.820E-01	265.7	8.954E-01	6.3E-06	4.3E-06
70	4.670E-02	230.7	7.051E-02	1.4E-07	8.6E-08
100	3.000E-04	210.2	5.000E-04	1.0E-09	4.3E-11

BIBLIOGRAPHY

Conrath, B. J. and I. Revah, A Review of Nonstatistical Techniques for the Estimation of Vertical Atmospheric Structure From Remote Infrared Measurements, M73-11588, Goddard Space Flight Center

Conrath, B. J., A Pdynomial Representation of Carbon Dioxide and Water Vapor Transmission, NESC 47, National Environmental Satellite Center, February 1969